Effiziente 3D-Interaktions- und Visualisierungstechniken für benutzer-zentrierte Modellierungssysteme

am Fachbereich Informatik

der Technischen Universität Darmstadt

genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

von

Dipl.-Inform. (FH) André Stork

aus Erzhausen

Referenten der Arbeit:

Prof. Dr.-Ing. J. L. Encarnação

Prof. Dr.-Ing. F.-L. Krause

Tag der Einreichung: 18. August 2000

Tag der mündlichen Prüfung: 29. September 2000

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fraunhofer-Institut für Graphische Datenverarbeitung in Darmstadt.

Mein besonderer Dank gilt Herr Prof. Dr. J. L. Encarnação, der mir die Möglichkeit gab, an seinem Institut zu arbeiten und durch vielfältige Unterstützung meiner Promotionsabsicht sowie meiner Forschungsaktivitäten zum Gelingen dieser Arbeit beigetragen hat.

Herrn Prof. Dr. F.-L. Krause möchte ich herzlich für die bereitwillige Übernahme der Korreferates sowie die rasche aber gründliche und wissenschaftlich-kritische Durchsicht des Manuskriptes danken.

Bei meinem Abteilungsleiter Herrn Dr. Rix möchte ich mich für die fachliche und moralische Unterstützung bedanken. Er hat mir nicht nur Möglichkeiten eröffnet, neue Ideen zu entwikkeln, sondern auch mit Partnern aus Forschung und Industrie zu diskutieren und umzusetzen.

Mein Dank gilt weiterhin vielen Kollegen und ehemaligen Kollegen, die durch Diskussion und konstruktive Kritik meine Forschungsaktivitäten ’mitgelenkt’ haben. Namentlich erwähnen möchte ich hier insbesondere Dr. U. Jasnoch, der mich von den ersten Tagen an in der Abteilung für Industrielle Anwendungen am IGD begleitet und wertvolle Diskussionsbeiträge geleistet hat, Dr. Kress, der als Maschinenbauingenieur sehr viel CAD-Know-How in die Diskussionen eingebracht hat sowie Hr. G. Brunetti, der mit seinen Forschungsaktivitäten auch Entwicklungen meinerseits stimuliert hat.

Ich möchte mich auch bei allen meinen Praktikanten und Diplomanden bedanken, die zur Realisierung meiner Ideen im System ARCADE beigetragen haben. In erster Linie ist hier Hr. M. Maidhof zu nennen, der später auch als Kollege ein wichtiger Diskussionspartner war. Von den betreuten Diplomanden möchte ich Herrn T. Richter, W. Schwert, R. Piecha und O. Schimpke an dieser Stelle nochmals meinen Dank aussprechen. Von den Praktikanten haben Herr I. Siedermann und L. Graßmann wichtige Beiträge geleistet.

Nicht zuletzt gebührt mein Dank allen anderen Mitarbeitern am Fraunhofer-IGD, die bei der Bewältigung der kleinen und großen Probleme des Alltags kollegial geholfen haben, hier insbesondere dem Sekretariat, dem INI-SC und der Videoecke.

Schließlich gilt meiner Familie ein ganz besonderes Dankeschön. Sie hat mir die Motivation, Kraft und Hilfe gegeben, die notwendig ist, um die mit einer solchen Arbeit verbundenen Belastungen zu bewältigen.

André Stork,

Erzhausen, im August 2000

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Softwaresysteme zur computer-unterstützten Konstruktion (CAD) haben sich in den vergangenen 30 Jahren von elektronischen Nachbildungen des Zeichenbrettes zu komplexen Werkzeugen für die 3-dimensionale Modellierung entwickelt [146]. 3D-CAD-Modelle sind zwingende Voraussetzung für Untersuchungen am virtuellen Modell und somit ein Schlüssel zur Verkürzung der Produktentwicklungszeit [147]. Zur Erstellung von 3D-Modellen verfolgen 3D-Modellierungssysteme einen technologie-zentrierten Ansatz, d.h. der Benutzer muß über z.T. konstruktionsfremde Systemkenntnisse verfügen, wie die CAD-Systemen zugrundeliegenden mathematischen Prinzipien, um mit ihnen arbeiten zu können.

Obwohl die 3D-Modellierung eine inhärent 3-dimensionale Aufgabe ist, was sich in Anbetracht traditioneller Modellierverfahren, wie z.B. dem Arbeiten mit Ton, leicht nachvollziehen läßt, setzen CAD-Systeme heute immer noch auf 2-dimensionale Ein- und Ausgabe. Der Einsatz von 2D-Eingabegeräten erfordert, daß 3D-Interaktionen als Sequenz von 2D-Eingaben nachgebildet werden müssen. Dazu muß der Konstrukteur wissen, wie das System mit den 2D-Eingaben verfährt und sich nach diesem richten. Das erfordert seinerseits ein entsprechendes mentales Modell und eine geistige Abbildungsleistung, was den effizienten Einsatz von 3D-CAD-Systemen insbesondere in bezug auf die geometrie-orientierten Funktionalitäten hemmt. Somit steckt gerade im Bereich des graphisch-interaktiven Dialogs noch Potential, um den Umgang mit 3D-Modellierungssystemen intuitiver und effizienter zu gestalten. Ausgabeseitig setzt sich das Manko heutiger CAD-Systeme fort, so erfolgt die Präsentation der 3D-Objekte auf einem 2D-Bildschirm und die verwendeten Visualisierungstechniken geben dem Betrachter nur unzureichende Form- und Tiefenhinweise, was die Orientierung und Navigation seitens des Benutzers sowie die Positionierung von Objekten erschwert.

Im Gegensatz zum technologie-orientierten Ansatz erfordert ein benutzer-zentrierter Ansatz die Beachtung der wahrnehmungspsychologischen sowie motorischen Fähigkeiten und Einschränkungen des Menschen. Benutzer-zentrierte Systeme zeichnen sich durch folgende Eigenschaften aus [51]:

- Sie bieten dem Benutzer eine effiziente Unterstützung bei der Bearbeitung seiner Aufgabe.
- Sie können flexibel eingesetzt werden und schränken den Benutzer nicht unnötig ein.
- Sie sind kontext-sensitiv und passen sich den sich ändernden Benutzeranforderungen an.
- Sie laden zum experimentellen Umgang ein und vereinfachen und beschleunigen Lernprozesse.
- Der Benutzer wird direkt miteinbezogen, sowohl was den Systementwurf als auch die Bedienung anbelangt.

Beim benutzer-zentrierten Ansatz steht der Anwender im Mittelpunkt; nicht er muß sich der Technologie unterordnen, sondern die Technologie unterstützt ihn auf natürliche und intuitive Weise bei der Bewältigung seiner Aufgaben. Eine benutzer-zentrierte Umgebung soll aber nicht nur einen Benutzer unterstützen, sondern auch dabei helfen, mit anderen Benutzern zu kooperieren. Das durch die Umgebung gebotene Maß an Immersion (dem Einbezogensein des Benutzers in die virtuelle Umgebung) und Kooperation soll an die zu bearbeitende Aufgabe adaptierbar sein.

Übertragen auf ein Modelliersystem implizieren diese Anforderungen benutzer-zentrierter Systeme eine integrierte Betrachtung von CAD, VR (Virtuelle Realität) - insbesondere was Visualisierung und 3D-Ein-/Ausgabegerätetechnologie anbelangt - und CSCW (computer supported collaborative work). Doch ein benutzer-zentriertes Modelliersystem ist mehr als die bloße Integration dieser Technologien; die o.g. Anforderungen müssen mit den Anforderungen von CAD in Einklang gebracht werden.

Im CAD steht die Präzision des geometrischen Modells mit seiner Produktsemantik im Vordergrund. CAD-Systeme zeichnen sich durch ihre umfangreiche Modellierfunktionalität aus. Im Vergleich zu VR-Systemen bieten sie jedoch kaum echte 3D-Interaktion und -Visualisierung. So kann festgestellt werden, daß das interne Modell oftmals das einzige 3-dimensionale an heutigen 3D-CAD-Systemen ist.

VR-Systeme hingegen sind nach Burdea [29] durch

- *immersion,*
- *interaction* (der Möglichkeit mit der Umgebung in Echtzeit zu interagieren) und
- *imagination* (der Illusion des Vorhandenseins manipulierbarer Objekte)

gekennzeichnet. Trotz der faszinierenden Möglichkeiten von VR, finden Ansätze, in VR zu modellieren, bislang kaum Einsatz und Akzeptanz. Dabei versprechen insbesondere 3D-Eingabegeräte einen intuitiven, natürlichen und direkt-manipulativen Umgang mit virtuellen Objekten. Sie reduzieren den Interaktionsaufwand und eröffnen neue Möglichkeiten der Mensch-Maschine-Kommunikation. Allerdings mangelt es an geeigneten Interaktionstechniken, die Modellierungs- und Konstruktionsaufgaben mit 3D-Eingabegeräten effizient unterstützen. Gegenüber den klassischen VR-Geräten 'Datenhandschuh' und 'Datenhelm' (head mounted display) lassen sich tischgebundene 3D-Eingabegeräte in den herkömmlichen Arbeitsplatz leicht integrieren und versprechen ein höheres Maß an Akzeptanz. Auch semi-immersive Präsentationsmedien, wie der Virtuelle Tisch (ein tisch-ähnliches Rückprojektionssystem), lassen sich gut in die gewohnte Arbeitsumgebung von Ingenieuren einbetten, die ja früher gewohnt waren, an Zeichenbrettern zu arbeiten.

Kooperationsunterstützung ist nicht nur aus einem benutzer-zentrierten Ansatz heraus gefordert, sondern auch aus dem Trend zu immer stärker räumlich verteilter Produktentwicklung [172]. Zwar existieren kommerzielle CSCW-Werkzeuge in unterschiedlicher Ausprägung, z.B. in Form von Videokonferenzsystemen und Sketching-Komponenten [162] sowie Application Sharing Tools [143], jedoch lassen diese Mechanismen keine unterschiedlichen Grade an Immersion in einer kooperativen Modellierungssitzung zu und erfüllen damit nicht die o.g. Anforderungen.

Durch ein benutzer-zentriertes, integratives Vorgehen können nicht nur die Beschränkungen heterogener CAD-, VR- und CSCW-Systeme gelöst, sondern es können auch die heute zwischen CAD und VR bestehenden Datenaustauschprobleme überwunden werden. Die Datenaustauschproblematik liegt darin begründet, daß CAD-Modelle zum Zwecke der schnellen Visualisierung

im VR-System aufbereitet werden müssen; sie werden trianguliert, reduziert und optimiert. Dadurch geht Genauigkeit und der Bezug zum ursprünglichen CAD-Modell weitestgehend verloren. Ein bidirektionaler Daten- und Informationsaustausch ist ohne weiteres nicht mehr möglich; der Weg von CAD nach VR wird zur Einbahnstraße.

Neben den Vorteilen für den Benutzer, bietet ein benutzer-zentriertes Modellierungssystem also auch Vorteile für die verteilte, virtuelle Produktentwicklung und überwindet Probleme des heutigen Einsatzes heterogener CAD- und VR-Systeme als auch CSCW-Werkzeuge.

1.2 Zielsetzung und Aufgabenstellung

Die Zielsetzung dieser Arbeit ist es, geeignete 3D-Interaktions- und Visualisierungstechniken für benutzer-zentrierte Modellierungssysteme zu entwickeln. Im Vordergrund steht dabei, 3D-Eingabegeräte für den Konstrukteur durch die Entwicklung neuartiger 3D-Interaktionstechniken und unterstützender Visualisierungstechniken nutzbringend einsetzbar zu machen.

Aus dieser Zielsetzung leiten sich folgende Fragestellungen ab:

- Wie muß ein benutzer-zentriertes Modellierungssystem beschaffen sein, um den Anforderungen von CAD, VR und CSCW zu genügen ?
    - Welche Systemarchitektur ist als Basis nötig, um diese Anforderungen zu befriedigen?
    - Welchen Einfluß hat die Forderung nach Direkt-Manipulation insbesondere auf die Benutzungsoberflächenkomponente eines benutzer-zentrierten Modellierungssystems?
    - Wie läßt sich Direkt-Manipulation und Telepräsenz für entfernt arbeitende Teilnehmer in einer kooperativen Sitzung realisieren?
- Wie sehen intuitive 3D-Interaktionstechniken für 3D-Eingabegeräte zur schnellen und präzisen Bauteilmodellierung und Zusammenbausimulation aus?
    - Welche unterstützenden Verfahren und Methoden sind nötig, um Effizienz und Präzision zu erreichen?
    - Lassen sich Modellieroperationen durch den Einsatz von 3D-Eingabe beschleunigen, automatisieren oder direkt-manipulativ gestalten?
    - Welche Visualisierungstechniken unterstützen eine effiziente Interaktion durch gesteigerte Form- und Tiefenwahrnehmung sowie verbesserte Orientierung und Navigation?
    - Welche neuen Möglichkeiten schaffen semi-immersive Umgebungen, wie der Virtuelle Tisch?
- Lassen sich die theoretischen Vorteile von 3D-Eingabe im Modellierungskontext nachweisen?
    - Sind mit der Einführung von 3D-Eingabegeräten und geeigneten Interaktionstechniken Vorteile bezüglich Intuitivität und Geschwindigkeit auch in der Detailkonstruktion zu realisieren?

Die vorliegenden Arbeit widmet sich in erster Linie der Volumenmodellierung nach der Definition von Pahl [111]. Die Feature-basierte, parametrische Modellierung [169] wird aus Sicht der Interaktion ebenfalls adressiert. In beiden Fällen spielt Präzision im graphisch-interaktiven Dialog eine Schlüsselrolle. Intuitiven Möglichkeiten zur Freiformflächen-Modellierung wird insbesondere am Virtuellen Tisch nachgegangen.

1.3 Zusammenfassung der Ergebnisse

Im Rahmen dieser Arbeit wurde basierend auf den allgemeinen Anforderungen an benutzerzentrierte Systeme und den spezifischen Anforderungen von CAD zunächst ein Architekturkonzept erarbeitet und prototypisch in Form des Systems ARCADE umgesetzt. ARCADE stellt ein kooperatives VR-CAD-System dar, das ein- und ausgabeseitig 3D unterstützt.

Durch Konzeptionierung und Realisierung neuartiger Methoden und Verfahren für die Interaktion mit 3D-Eingabegeräten, konnte das Ziel des effizienten und präzisen Modellierens mit solchen Geräten umgesetzt werden. Die wichtigsten der erarbeiteten Verfahren sind:

- Semantisch-korrekte, typbezogene Interaktionsmethoden zur Objekterzeugung in 3D.
- Gesten-basierte Manipulation mittels eines neuen Interaktionsparadigmas, der Topologie-basierten eingeschränkten Modifikationstechnik (TCBM).
- Diskretisierungsmöglichkeiten für echte 3D-Interaktionen, die ein präzises und effizientes Konstruieren in einem rein graphisch-interaktiven Dialog mit 3D-Eingabegeräten erlauben.

Als essentiell hat sich die Notwendigkeit eines effizienten 3D-Pickings und -Snappings erwiesen. Basierend auf dieser Erkenntnis wurde ein Verfahren zum schnellen Picking und Snapping des graphischen 3D-Echos auf die präzise Geometrie des CAD-Modells entwickelt.

Der Einsatz von 3D-Eingabegeräten hat neue direkt-manipulative Modellieroperationen, wie implizite Boolsche Operationen und direkt-manipulatives Sweeping ermöglicht, die den Konstruktionsprozeß weiter beschleunigen.

Zur Unterstützung der Zusammenbausimulation wurde ein Ansatz konzipiert und prototypisch umgesetzt, der Assembly-Attribute im Modellierungsprozeß automatisch erzeugt und basierend auf dieser Attributinformation, dem Benutzer beim Zusammenbau virtueller Teilmodelle mit 3D-Eingabegeräten Hilfestellung bietet.

Hinsichtlich des Ziels, dem Benutzer aufschlußreiche Form- und Tiefenhinweise zu bieten und ihm die im Gange befindliche Interaktion geeignet darzustellen, wurden Visualisierungstechniken erarbeitet, umgesetzt und in bezug auf ihre unterstützende Wirkung auf 3D-Interaktionen hin evaluiert - letzteres insbesondere für stereoskopische Darstellung und Schattenwurf. Darstellungstechniken, die dem Benutzer die stattfindende Interaktion sowie die Diskretisierung und die erlaubten Modifikationsmöglichkeiten anzeigen, wurden in Benutzertests von der Mehrheit der Testpersonen als sehr gut und wünschenswert auch für das ihnen vertraute CAD-System beurteilt.

Ein Vergleich mit kommerziellen Systemen im Rahmen eines Benutzertests, bei dem die *expert performance* für eine Modellierungsaufgabe, bei der Präzision gefordert war, erhoben wurde, hat ergeben, daß sich mit 3D-Eingabegeräten in Kombination mit den entwickelten 3D-Interaktions- und Visualisierungstechniken ein Geschwindigkeitsvorteil von einem Faktor 2 bis 4 realisieren läßt.

1.4 Gliederung

Den Ausgangspunkt dieser Arbeit bildet der in Kapitel 2 dargelegte Stand der Technik. Dabei werden neben Systemarchitekturen insbesondere 3D-Interaktionstechniken für 2D- und 3D-Eingabegeräte unter dem Aspekt der 3D-Modellierung diskutiert. Es werden sowohl ein- als auch zweihändige Ansätze vorgestellt. Darüber hinaus werden 3D-Ein- und Ausgabegeräte bezüglich

ihrer Eignung für 3D-Modellierungsanwendungen in verschiedenen Umgebungen gegenübergestellt. Eine Betrachtung des menschlichen Sehens und dessen Bedeutung für Visualisierung und Interaktion schließt das Kapitel ab.

Basierend auf den Unzulänglichkeiten des Standes der Technik, den allgemeinen Anforderungen an benutzer-zentrierte Systeme sowie den speziellen Anforderungen der 3D-Modellierung wird in Kapitel 3 eine Systemarchitektur für benutzer-zentrierte 3D-Modellierungssysteme konzipiert. Schwerpunkt dieser Architektur bilden die Komponenten des Benutzungsoberflächensystems. Der impliziten Forderung nach Kooperationsunterstützung benutzer-zentrierter Systeme wird mit einer Netzwerkkomponente, die inhärenter Bestandteil des Architekturkonzeptes ist, begegnet.

Kapitel 4 greift ebenfalls die Mankos des heutigen Standes der Technik auf und widmet sich den unter besonderer Berücksichtigung der Modellierung entwickelten neuartigen Interaktionstechniken zur schnellen und präzisen Bauteilkonstruktion und Zusammenbausimulation mittels 3D-Eingabegeräten. Neben den Interaktionstechniken werden in Kapitel 4 die unterstützenden Visualisierungstechniken sowie Methoden und Verfahren eingeführt, die effizientes und präzises Arbeiten mit 3D-Eingabegeräten in der Modellierung ermöglichen.

In Kapitel 5 erfolgt die Evaluierung der entwickelten Interaktions- und Visualisierungstechniken. Die Interaktionstechniken zur schnellen und präzisen Bauteilmodellierung werden qualitativ mit bekannten Ansätzen aus der Forschung und Entwicklung als auch quantitativ mit kommerziellen CAD-Systemen verglichen. Darüber hinaus werden verschiedene Experimente zur Wechselwirkung zwischen Visualisierung und Interaktion beschrieben.

Abschließend werden in Kapitel 6 die Ergebnisse der Arbeit zusammengefaßt und ein Ausblick auf weiterführende Arbeiten gegeben.

2 Stand der Technik

Im Hinblick auf die Themenstellung der vorliegenden Arbeit sind verschiedene Forschungsbereiche von Bedeutung. Dabei spielen 3D-Interaktionstechniken die Hauptrolle. Visualisierungstechniken sollten sich die wahrnehmungspsychologischen Erkenntnisse über den menschlichen Wahrnehmungsapparat zunutze machen. Darum werden in diesem Kapitel Grundlagen zur Form- und Tiefenwahrnehmung dargestelt. Darüber hinaus sind Systemarchitekturen von CAD- und VR-Systemen für die Konzipierung eines benutzer-zentrierten Modellierungssystems von Belang. Dieses Kapitel gibt einen Überblick über den aktuellen Stand der Technik dieser Gebiete sowie über die Ein- und Ausgabegerätetechnologie und bewertet diese im Hinblick auf die Anforderungen der Modellierung.

2.1 3D-Eingabegeräte

3D-Eingabegeräte wurden entwickelt, um dem Benutzer ein möglichst natürliches und intuitives Interagieren in 3-dimensionalen, computer-generierten Szenen zu ermöglichen. 3D-Eingabegeräte ermöglichen es, echte 3D-Interaktionen durchzuführen und bieten somit prinzipielle Vorteile gegenüber 2D-Eingabegeräten, mit denen 3D-Interaktionen als Sequenzen von 2D-Interaktionen nachempfunden werden müssen. 3D-Eingabegeräte stellen dem Benutzer i.d.R. sechs Freiheitsgrade zur Verfügung, nämlich drei für die Positionierung und drei für die Orientierung im Raum. 3D-Eingabegeräte lassen sich in die drei Kategorien einteilen:

- ”fliegende” Geräte,
- gesten-gesteuerte Geräte und
- ”tisch-gebundene” Geräte [74].

Die drei Kategorien werden im folgenden kurz beleuchtet, je ein relevanter Vertreter vorgestellt und unter dem Aspekt der Eignung für Modellierungsaufgaben bewertet. Dazu werden zunächst die Anforderungen, die CAD an diese Geräte stellt, diskutiert.

2.1.1 Anforderungen an 3D-Eingabegeräte aus der Sicht von CAD

Innerhalb dieses Abschnitts werden die Anforderungen formuliert, die 3D-CAD an 3D-Eingabegeräte stellt. Bei der Anforderungsanalyse und Bewertung wurde auf Aussagen in [174], [144], [94] und [57] zurückgegriffen und diese um Aspekte der 3D-Modellierung [149] ergänzt.

Die Entwicklung von 3D-Eingabegeräten wurde primär im Bereich der Virtuellen Realität vorangetrieben. Das Ziel von VR-Systemen besteht in erster Linie darin, den Benutzer in eine künstliche, computer-generierte Welt, eintauchen zu lassen. Der Benutzer (in VR auch ’cybernaut’ genannt) kann sich mittels multidimensionaler Interaktionstechniken durch die virtuelle Umgebung bewegen und Objekte manipulieren. Er kann mit diesen Objekten in ”gewohnter”

Weise interagieren, d.h. sie greifen, bewegen und wieder loslassen. Dabei sind die Objekte beim Eintritt in die virtuelle Welt i.d.R. bereits vorhanden (unter den Anwendungen in VR stellt die Modellierung von Objekten eine Ausnahme dar). Eine kompakte Einführung in Methoden, Techniken und Anwendungsgebiete der Virtuellen Realität bieten [10] und [11].

Modellierungsfunktionalität, wie das Erzeugen und Modifizieren von Objekten, wird von VR-Systemen nur in Ausnahmen unterstützt. Im Unterschied zu VR-Systemen sind dies Kernfunktionalitäten eines Modellierungssystems. Mit einem Modellierungssystem muß es möglich sein, Objekte exakt zu erzeugen und zu modifizieren, d.h. Objekte müssen genau dimensioniert werden können. Konstrukteure arbeiten mit ihrem CAD-System täglich oft mehrere Stunden, daher werden an die Gebrauchstauglichkeit der Eingabegeräte andere Anforderungen gestellt als in VR. 3D-Eingabegeräte für Modellierungsaufgaben sollten folgenden Anforderungen genügen:

- Sie sollten präzise sein, d.h. Raumpunkte müssen exakt bestimmt werden können.
- Ihr Aktionsradius darf nicht beschränkt sein, da der Konstruktionsraum prinzipiell unbegrenzt ist.
- Rotationen um beliebig große Rotationswinkel müssen einfach ausführbar sein.
- Sie müssen ergonomisch sein, um eine Nutzung über einen längeren Zeitraum zu ermöglichen.
- Sie dürfen den Benutzer nicht einschränken, so daß dieser auch weiterhin 2D-Interaktionen bzw. Tastatureingaben ausführen kann.
- Sie sollten dem Benutzer ein entsprechendes Feedback bei der Kraftanwendung übermitteln, um die Bedienung zu erleichtern.
- Sie sollten im Vergleich zu den Kosten eines CAD-Arbeitsplatzes preiswert sein.

2.1.2 Die SpaceMouse - ein tischgebundes 3D-Eingabegerät

Die SpaceMouse [98] ist ein tisch-gebundenes 3D-Eingabegerät, das sechs Freiheitsgrade zur Verfügung stellt (siehe Abbildung 1). Die SpaceMouse bietet neun frei programmierbare Tasten und eine bewegliche Kappe, die die sechs Freiheitsgrade realisiert. Um die Kappe auszulenken, muß eine gewisse Kraft aufgewendet werden. Die Kappe der SpaceMouse reagiert darauf mit einer Bewegung innerhalb enger Grenzen (+/- 1,5 mm für die Translation und +/- 4 Grad für die Rotation). Diese Form der Rückkopplung gibt dem Benutzer ansatzweise ein Gefühl dafür, wie weit er die Kappe ausgelenkt hat.



Abbildung 1: Die SpaceMouse [98]

Bei den von der SpaceMouse gelieferten Werten handelt es sich um relative Positions- und Rotationswerte. Die Werte geben also die Translations- bzw. Rotationsänderungen im letzten Zeitin-

tervall an. Wird die Kappe der SpaceMouse gleichbleibend in eine Richtung gedrückt, so bewegt sich das korrespondierende graphische Objekt entsprechend. Je größer die Auslenkung der Kappe der SpaceMouse, desto größer ist der gelieferte Translations-/Rotationswert, d.h. desto schneller bewegt sich das korrespondierende Objekt.

2.1.3 Der Polhemus-Sensor - ein fliegendes 3D-Eingabegerät

”Fliegende” 3D-Eingabegeräte sind Geräte, die an der Hand des Benutzers befestigt oder in der Hand gehalten und im Raum bewegt werden. Die Position und Orientierung des Eingabegerätes bzw. der Hand im Raum wird z.B. unter Zuhilfenahme eines elektromagnetischen Feldes bestimmt. Im folgenden wird der Polhemus-Sensor [1] beschrieben, der eines der meistgenutzten fliegenden Systeme repräsentiert.

Der Polhemus Sensor, im folgenden auch nur kurz Polhemus genannt, ist ein 3D-Positions- und Orientierungssensor, der eigenständig oder in Verbindung mit anderen Eingabegeräten, z.B. dem Datenhandschuh (siehe Kapitel 2.1.4), genutzt wird. Zur Bestimmung der 3D-Position und -Orientierung benutzt der Polhemus eine elektromagnetische Kopplung zwischen einem Sender und einem Empfänger. Der Sender wird an einer festen Position im Raum angebracht, während der Polhemus-Sensor, der den eigentlichen ’Empfänger’ darstellt, frei im Raum bewegt werden kann. Die absolute Position und Orientierung des Sensors in dem von dem Sender ausgestrahlten elektromagnetischen Feld wird zyklisch bestimmt. Zur Auswertung der Position und Orientierung ist der Sensor über ein Kabel mit dem Rechner bzw. einer Schnittstellen-Station verbunden, die die Positionsbestimmung durchführt.

2.1.4 Der Datenhandschuh - ein gesten-basiertes Eingabegerät

Der Datenhandschuh [10] bildet in Verbindung mit fliegenden 3D-Eingabegeräten die klassische Eingabetechnologie für VR-Systeme. Der Datenhandschuh erfaßt dabei die Hand- und Fingerbewegungen seines Benutzers. Aus der Stellung der Finger können Handgesten interpretiert werden, die dann entsprechende Interaktionen auslösen. Typischerweise wird der Datenhandschuh in Kombination mit einem Positions- und Orientierungs-Sensor (z.B. dem Polhemus) benutzt, der die absolute Position und Orientierung der Hand im Raum bestimmt.

Die Erkennung der Fingerstellung seitens des Handschuhs geschieht zumeist über ein optisches Meßsystem. Der Handschuh ist dabei an der Handoberseite mit dünnen Glasfaserkabeln bestückt, die über die Finger gespannt sind. Ein Ende jedes Glasfaserkabels ist mit einer Leuchtdiode versehen, das andere Ende mit einem Phototransistor, der das ankommende Licht der Diode mißt. Die Krümmung eines Fingergelenks reduziert die ankommende Lichtenergie, was Aufschluß über die Stellung des Fingers gibt. Gebräuchliche Datenhandschuhe registrieren bis zu 16 unterschiedliche Fingerstellungen.

In VR-Systemen werden Bewegungen der Hand des Benutzers häufig auf ein graphisches Echo in Form einer virtuellen Hand übertragen. Die virtuelle Hand korrespondiert mit der realen Bewegung, was eine natürliche und intuitive Navigation und Manipulation ermöglicht. Innerhalb der 3D-Szene werden die Gesten und Bewegungen des Benutzers von der virtuellen Hand nachvollzogen. Über den gewohnten Bewegungsablauf kann der Benutzer Objekte greifen, bewegen und wieder loslassen, die Interaktion verläuft somit sehr intuitiv. Der Datenhandschuh wird meistens in immersiven Anwendungen eingesetzt, also in Anwendungen, in denen der Benutzer völlig in die Darstellung der computer-generierten Szene einbezogen wird. Vollständige

Immersion wird durch kopfgebundene Darstellungssysteme (head mounted displays) oder CAVEs [42] erreicht.

2.1.5 Sonstige 3D-Eingabegeräte

Neben den hier näher beschriebenen Eingabegeräten bietet der Markt eine stetig steigende Anzahl anderer 3D-Eingabegeräte. Neben der SpaceMouse und dem SpaceBall sowie den diversen Datenhandschuhen beginnen sich 3D-Eingabegeräte mit taktilem Feedback und Geräte mit Kraftrückkopplung (Phantom [181]) oder auch kabellose 3D-Eingabegeräte [112] zu etablieren. Darüber hinaus werden auch Eingabegeräte für spezielle Anwendungsfälle, wie das Deformieren von Objekten, entwickelt [103]. Andere Wissenschaftler versuchen die Eingabemöglichkeiten ohne explizites Eingabegerät zu verbessern; hier sind in erster Linie Video-basierte Systeme zu nennen, die spezielle Markierungen aufnehmen und mit Hilfe von Bilderkennungsverfahren die Position dieser Punkte im Raum bestimmen [137]. Eine Übersicht über Eingabegerätetechnologie ist unter [32] zu finden.

2.1.6 Bewertung und Auswahl von 3D-Eingabegeräten

Im folgenden werden die vorgestellten Eingabegeräte bezüglich ihrer Eignung für den Modellierungsprozeß am herkömmlichen Arbeitsplatz bewertet. Anzumerken ist, daß die Umgebung, in der ein 3D-Eingabegerät eingesetzt werden soll, die Prioritäten bei den o.g. Anforderungen maßgeblich beeinflußt und somit die Art des bevorzugten Eingabegerätes mitbestimmt. So erscheinen für den herkömmlichen Arbeitsplatz tischgebundene Geräte geeigneter. Der Virtuelle Tisch (vgl. Kapitel 2.3.3) legt hingegen den Einsatz fliegender Geräte nahe. Auch wird durch den Virtuellen Tisch der Zugriff auf eine normale Tastatur erschwert und andere Eingabemodi (z.B. Sprachkommandos) gewinnen an Bedeutung.

Die SpaceMouse weist eine sehr gute Ergonomie auf. Ihre Kappe ist vergleichsweise flach und kann von der Hand bequem umfaßt werden, so daß kaum Ermüdungserscheinungen auftreten. Der Meßmechanismus der SpaceMouse ist einfach, unanfällig gegenüber Verschleißerscheinungen und von hoher Präzision. Darüber hinaus ist die SpaceMouse im Vergleich zu magnetischen Trackern relativ preiswert. Somit erfüllt sie die genannten Anforderungen an ein 3D-Eingabegerät für einen herkömmlichen CAD-Arbeitsplatz in vorbildlicher Weise. Ein Nachteil tisch-gebundener Geräte ist, daß sie nur indirekte 3D-Interaktionen (vgl. Kapitel 2.2) erlauben.

Der Vorteil fliegender Eingabegeräte, nämlich die Möglichkeit zur direkten 3D-Interaktion (vgl. Kapitel 2.2) ist zugleich ihr größter Nachteil: Die Tatsache, daß sie mit der Hand und dem Arm vor dem Körper geführt werden müssen, führt schnell zur Ermüdung. Ein weiterer Nachteil ist die i.d.R. notwendige Verkabelung des Benutzers. Die Geräte sind nicht für das Arbeiten über einen längeren Zeitraum geeignet. Außerdem liefern fliegende Geräte meist absolute Raumpositionen, d.h. der Interaktionsraum ist vorbestimmt. Es muß ein ”Einfriermechanismus” realisiert werden, soll die Hand ohne Beeinflussung der graphischen Echos bewegt werden können, um das graphische Echo über den physikalischen Tracking-Bereich hinaus verschieben zu können.

Elektromagnetische Tracking-Systeme weisen im Hinblick auf die Anforderungen der 3D-Modellierung folgende Nachteile auf:

- Der Sensor ist störanfällig in bezug auf den Einfluß eines Fremdfeldes, so können sich schon naheliegende metallische Objekte störend auswirken.
- Die räumliche Auflösung ist begrenzt, so daß Positionen nicht beliebig präzise erfaßt bzw. berechnet werden können.

- Die Reaktion des graphischen Feedbacks erfolgt zum Teil zeitlich verzögert (infolge der Positionsbestimmung und Rauschunterdrückung).

Der Datenhandschuh erlaubt es nicht, auf schnelle und einfache Art und Weise das Eingabegerät zu wechseln oder auch nur das Eingabegerät ’loszulassen’, um z.B. kurz etwas zu notieren. Er läßt sich also kaum in den typischen Arbeitsablauf eines Konstrukteurs einbetten. Der Einsatz eines Datenhandschuhs über einen längeren Zeitraum ist daher für Modellierungsaufgaben nicht sinnvoll. Im folgenden sind weitere Gründe aufgeführt, die gegen die Verwendung eines Datenhandschuhs innerhalb eines 3D-CAD-Systems sprechen:

- Typische CAD-Interaktionen, wie z.B. die Rotation um einen großen Winkel, sind mit dem Datenhandschuh nur umständlich ausführbar, da die Hand natürlichen Bewegungseinschränkungen unterliegt; bei größeren Winkeln wird ein Umgreifen nötig.
- Der Datenhandschuh muß individuell auf die Hand des Benutzers kalibriert werden, um befriedigende Resultate zu liefern.
- Der Handschuh liegt sehr eng an, was sich bei längerem Tragen unangenehm auswirkt.
- Das Fehlen von Kraftrückkopplung wird als störend empfunden; der Benutzer erhält beim Greifen i.d.R: kein taktiles Feedback.
- Für Rechts- und Linkshänder werden unterschiedliche Geräte benötigt.

Einige der genannten Probleme werden durch neue Entwicklungen, wie z.B. dem Einsatz neuronaler Netze zur Gestenerkennung [23] oder durch Handschuhe mit taktilem Feedback (CyberTouch [182]) und Kraftrückkopplung (CyberGrasp [182]), gelöst bzw. gemildert. Die Erfahrungen aus den im Rahmen dieser Arbeit erfolgten Benutzertests mit Konstrukteuren lassen jedoch den Schluß zu, daß der Einsatz eines Datenhandschuhs bei der täglichen Arbeit inakzeptabel ist. Abschließend erfolgt eine tabellarische Gegenüberstellung der präsentierten 3D-Eingabegeräte hinsichtlich der formulierten Anforderungen.

Tabelle 1: Gegenüberstellung verschiedener 3D-Eingabegeräte bezüglich ihrer Eignung für den Einsatz in Modellierungsanwendungen am herkömmlichen Arbeitsplatz [99]

	Polhemus (evtl. gekoppelt mit Maus -> ”Bat”)	**Datenhandschuh**	**SpaceMouse**
Präzision	-	-	++
unbegrenzter Aktionsraum	- -	- -	++
unbegrenzter Rotationswinkel	-	- -	++
Ergonomie	- -	- -	++
2D-Eingabe weiterhin möglich	-	- -	++
Feedback bei Kraftanwendung	-	-	+
intuitive Interaktion	++	++	+
Preis	-	- -	++

Legende: ++ sehr gut, + gut, - schlecht / hoch, - - sehr schlecht / sehr hoch

2.2 3D-Interaktion

Unter 3D-Interaktion wird das Agieren des Benutzers in computer-generierten Szenen verstanden. Dabei lassen sich verschiedene Basis(inter)aktionen unterscheiden:

- Das **Navigieren**, d.h. das zielgerichtete Bewegen im Raum realisiert durch Veränderungen von Kameraposition und -orientierung.
- Das **Positionieren**, d.h. das Plazieren eines Objektes an einer bestimmten Raumposition.
- Das **Orientieren**, d.h. das Rotieren eines Objektes im Raum.
- Das **Selektieren** bzw. **Deselektieren**, d.h. das Auswählen eines Punktes oder Objektes im Raum.

Häufig werden Positionieren und Orientieren zusammengefaßt, da das die beiden Aufgaben sind, die mit einem 3D-Eingabegerät, das sechs Freiheitsgrade besitzt, simultan durchgeführt werden können.

Aus diesen Basisinteraktionen lassen sich höherwertige Aktionen, wie das Bewegen von Objekten zu einem Ort, ableiten. Dazu wird zunächst das graphische Echo, das den Benutzer in der Szene repräsentiert (Cursor oder virtuelle Hand) zu dem Objekt geführt, welches bewegt werden soll und das Objekt selektiert. Das Objekt folgt der sich anschließenden Hand- bzw. Cursorbewegung und wird schließlich deselektiert bzw. losgelassen.

Im Rahmen der Interaktion ist die **Direktmanipulation** (*direct manipulation*) [141] ein wichtiger Begriff und eine der Grundforderungen, die an graphisch-interaktive Systeme gestellt werden. Direkte Manipulation ist gegeben, wenn Aktionen unmittelbar zu einer sichtbaren Wirkung führen. Im Idealfall ist das Zeitintervall zwischen Aktion und visueller Rückkopplung nicht länger als 1/20 Sekunde. Dies entspricht der Bildwiederholrate, die gerade ausreicht, damit der Benutzer den Eindruck einer fließenden Bewegung bekommt. Dieser Idealzustand ist oft nicht erfüllbar, so haben Untersuchungen ergeben, daß eine Zeitverzögerung von bis zu einer Sekunde von Benutzern in bestimmten Anwendungsfällen toleriert wird. Anzustreben ist eine Verzögerung von maximal einer zehntel Sekunde.

Ein weiterer wichtiger Begriff ist die sogenannte **Reiz-Reaktions-Korrespondenz** (engl. *stimulus response correspondence* oder auch *stimulus response compatibility)* [115], [59]. Die Reiz-Reaktions-Korrespondenz ist eine der grundlegenden Anforderungen an intuitive Mensch-Maschine-Schnittstellen.

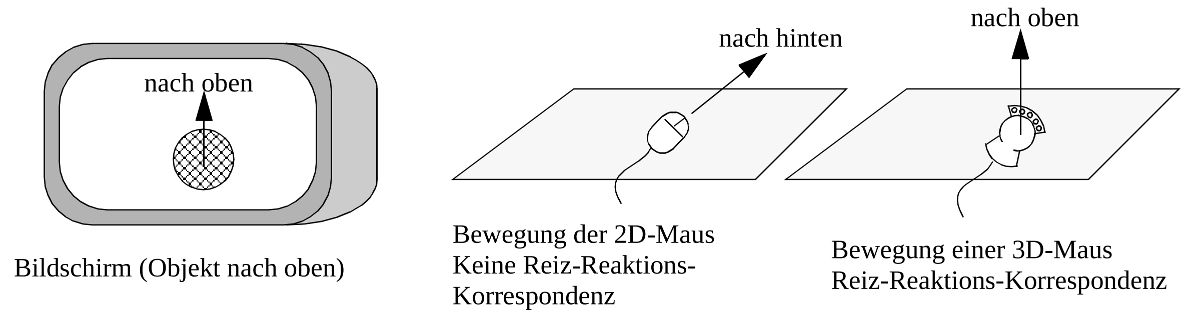


Abbildung 2: Prinzip der Reiz-Reaktions-Korrespondenz

Eine Korrespondenz von Reiz und Reaktion liegt vor, wenn auf einen Reiz, z.B. eine Handbewegung nach links, eine Objektbewegung in dieselbe Richtung erfolgt. Bei allen Interaktionen mit realen Gegenständen in der realen Welt korrespondiert Reiz und Reaktion. Beim Arbeiten mit 2D-

Benutzungsoberflächen und der 2D-Maus hingegen liegt keine Reiz-Reaktions-Korrespondenz vor, denn der Mauszeiger bewegt sich auf dem Bildschirm nach oben, wenn die Maus vom Benutzer weg ’nach hinten’ bewegt wird (siehe Abbildung 2). Liegt keine Reiz-Reaktions-Korrespondenz vor, so ist eine mentale Abbildung (*mental mapping*) seitens des Benutzers notwendig, um zur gewünschten Aktion die korrespondierende Bewegung auszuführen. Reiz-Reaktions-Korrespondenz für 3-dimensionale Aufgaben, wie die Volumenmodellierung, kann nur mit 3D-Eingabegeräten erzielt werden und minimiert Aufwand sowie Umfang mentaler Abbildungsvorgänge.

Der Begriff der **direkten bzw. indirekten 3D-Interaktion** ist im Verlauf der Arbeit ebenfalls von Bedeutung und soll deshalb an dieser Stelle definiert werden. Direkte Interaktion liegt vor, wenn die Position der Hand des Benutzers mit der Position des graphischen Echos des Eingabegerätes übereinstimmt. Indirekte Interaktion liegt vor, wenn die Hand des Benutzers sich nicht an der Position des Echos des Eingabegerätes befindet. Die herkömmliche Arbeitsweise mit der 2D-Maus ist eine Form der indirekten Interaktion, da sich die Hand nicht an der Stelle des Mauszeigers befindet. Das Arbeiten mit einem Datenhandschuh in einer immersiven Umgebung ist hingegen meist von direkter Interaktion geprägt.

2.2.1 3D-Interaktion mit 2D-Eingabegeräten

Obwohl die 3D-Interaktion mit 3D-Eingabegeräten in dieser Arbeit eine vorrangige Rolle einnimmt, werden zunächst die wichtigsten Ansätze zur 3D-Interaktion mit 2D-Eingabegeräten vorgestellt, um die Möglichkeiten aber auch die Grenzen von 3D-Interaktion mit 2D-Eingabegeräten und die notwendigen Interaktionstechniken zu verdeutlichen.

2.2.1.1 Allgemeine Techniken

Bier [15], [16], [17] führt graphische Primitive, sog. *skitters* und *jacks*, zur interaktiven Positionierung von Objekten in 3D ein. Durch die Kombination von interaktiver Plazierung und Eingabe von Werten lassen sich Objekte präzise aneinander ausrichten. Die Möglichkeiten zur Direktmanipulation sind beschränkt, da die eigentliche Objektmanipulation nach der alpha-numerischen Werteingabe erfolgt.

Nielson und Olson [107] präsentieren ein Verfahren, das die 2-dimensionale Bewegung der Maus auf eine 3-dimensionale Bewegung abbildet, in dem sie die Bewegungsebene der Maus in Zonen unterteilen und Bewegungen innerhalb verschiedener Zonen auf unterschiedliche Achsen im Raum abbilden (siehe Abbildung 3). Anstatt mit Zonen, läßt sich eine Bewegung eines Objektes im Raum auch mit der 2D-Maus in Kombination mit Tasteneingabe ereichen [58]. Der von Nielson und Olson vorgeschlagene Mechanismus erlaubt Translationen, Rotationen und Skalierungen. Verfahren zur Objekterzeugung werden nicht vorgestellt. Nielson und Olson zeigen, daß 3D-Interaktionen - vergleichsweise umständlich - auch ohne 3D-Eingabegeräte möglich sind. Die Reiz-Reaktions-Korrespondenz in 3D ist bei ihrem Ansatz nicht gegeben. In ihrer Arbeit taucht die Idee auf, einen *full space cursor* einzusetzen. Der *full space cursor* ist ein graphisches Echo, das sich als 3D-Fadenkreuz durch den Raum erstreckt. Auch die Möglichkeit, durch Fußpunkte die Enden des *full space cursors* zu markieren, wird in ihrer Arbeit erstmals erwähnt.

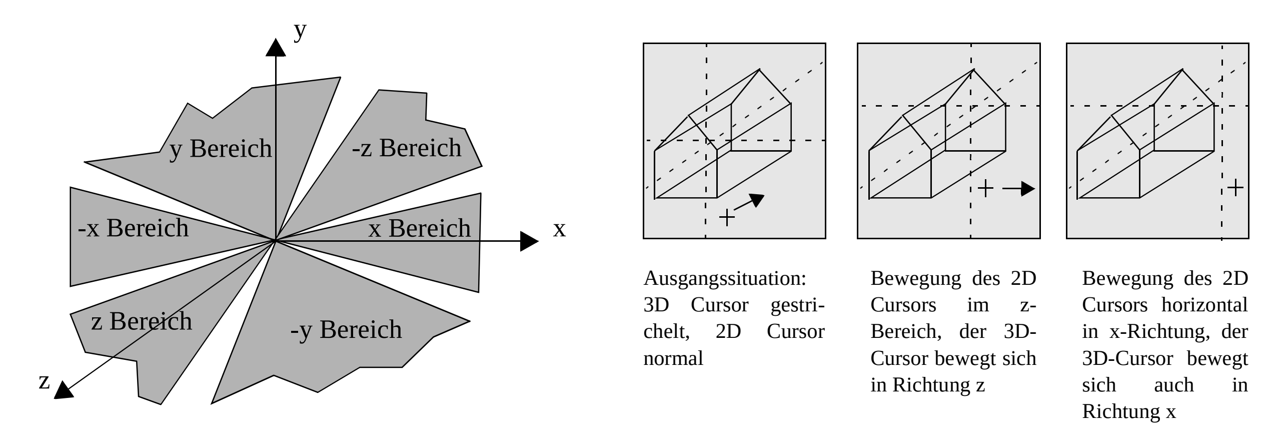


Abbildung 3: 3D-Cursorsteuerung mit 2D-Maus nach Nielson und Olson [107]

Chen et al. [34] untersuchten das interaktive Rotieren in 3D mit 2D-Eingabegeräten und entwikkelten die virtuelle Kugel (*virtual sphere*) als Rotationsmetapher. Bewegungen der Maus werden als Bewegungen entlang der Längen- und Breitengrade einer virtuellen Kugel interpretiert, die das zu rotierende Objekt bzw. die zu rotierende Szene scheinbar umgibt (siehe Abbildung 4). Somit kann das Objekt beliebig in 3D gedreht werden. Diese Metapher ist mittlerweile stark verbreitet, da sie von Benutzern als intuitiv empfunden wird.

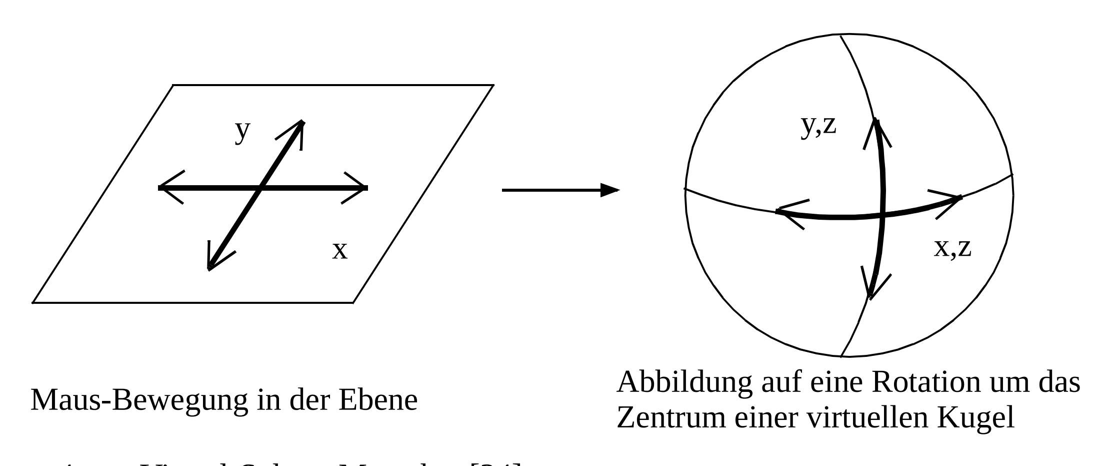


Abbildung 4: Virtual-Sphere-Metapher [34]

Galyean et al. [66] verallgemeinern die aus 2D-Systemen bekannten '*handles*' auf 3D. Handles (auch 3D-Widgets genannt) sind kleine graphische Primitive, die ein Objekt umgeben. Das Bewegen eines Handles mit der Maus führt zur Transformation des korrespondierenden Objektes (siehe Abbildung 5). Van Emmerik [167] und Rappoport [117] stellen auf CAD-Anwendungen spezialisierte 3D-Widgets vor. Die Idee ist heute in vielen Anwendungen und Graphiksystemen, wie z.B. OpenInventor [178] umgesetzt. Als Synonym wird auch die Bezeichnung Manipulatoren oder 3D-Widgets benutzt.

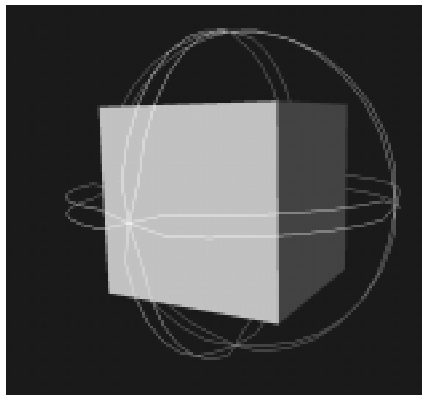
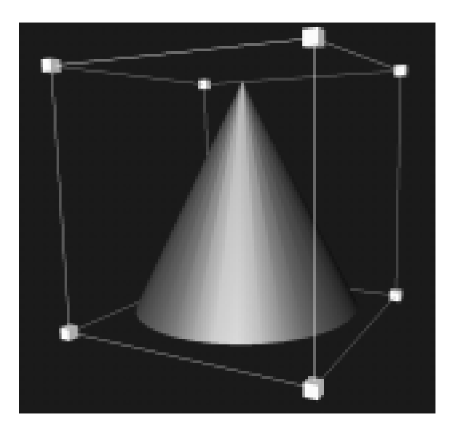
 

Abbildung 5: 3D-Widgets zur Rotation (links) und zur allgemeinen Transformation (rechts) [178]

3D-Widgets sind gut geeignet, um eine 2D-Aktion in eine 3D-Interaktion zu überführen, aber sie verfügen, wie im Rahmen dieser Arbeit durchgeführte Experimente zeigen, über eine Reihe von Nachteilen:

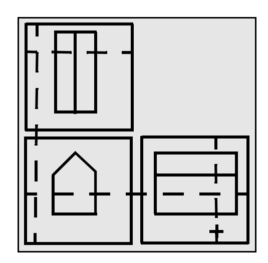
- Sie überfrachten die Szene mit zusätzlicher Geometrie, die Objekte verdecken kann.
- Die Interaktion läuft indirekt über die Handles ab, nicht über die eigentlich zu manipulierenden Objekte.
- Sie unterstützen oft nur unpräzise Interaktionen; es mangelt an Möglichkeiten zur Diskretisierung.
- Die Reiz-Reaktions-Korrespondenz ist nicht gewährleistet; es kommt zu nicht-erwartungskonformen Objektbewegungen.
- Dem Betrachter näher gelegene Handles können weiter entfernt liegende Handles verdecken. Der Benutzer muß die Szene drehen, um das ’richtige’ Handle picken zu können.

2.2.1.2 Techniken zur Modellierung von 3D-Objekten

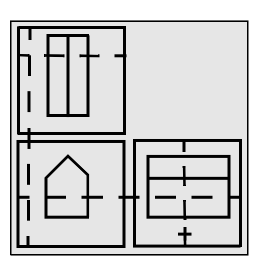
In heutigen 3D-CAD-Systemen kommt zumeist eines der folgenden Interaktionskonzepte zum Einsatz:

- Das Konzept der multiplen Ansichten:
  Drei orthogonale Ansichten des 3D-Modells werden nebeneinander dargestellt [88],[72]. Diese Darstellungsform verliert zunehmend an Bedeutung. Eine 3D-Position muß dabei durch zweimaliges Picken in zwei unterschiedlichen Fenstern angegeben werden (siehe Abbildung 6).
- Das Konzept der Arbeitsebene:
  Es erweitert den obigen Ansatz, in dem eine beliebige Ebene, z.B. die Fläche eines existierenden Körpers, ausgewählt werden kann, um darauf weiter zu konstruieren. Bei allen Eingaben wird der Schnittpunkt zwischen Pickstrahl - definiert durch Kamera und Mausposition - und der gewählten Ebene und somit eine Position in 3D bestimmt. Bei der Objekterzeugung auf vorhandenen Objekten wird der Schnitt mit der Arbeitsebene durch den Schnitt mit dem entsprechenden Objekt ersetzt.

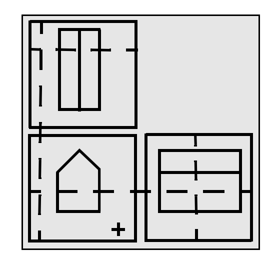
- Das sequentielle Abbilden der Mausbewegung auf Parameter geometrischer Primitive: Dieses Konzept kommt oft bei CSG-Ansätzen zum Einsatz. Relativ zu einem definierbaren Ursprung werden die Mausbewegungen sukzessive auf Objektparameter abgebildet. Die Definition eines Parameters muß durch Klicken abgeschlossen werden. Die Reihenfolge der Parametereingabe ist vorgegeben [56].



Um den gestrichelten 3D Cursor zu bewegen, wird er mit dem 2D-Mauszeiger gepickt



Bewegungen des Mauszeigers führen nun zur Translation des 3D Cursors in der Ebene.



Nach Loslassen des Mausknopfs kann der 2D Cursor wieder frei bewegt werden, ohne daß der 3D Cursor folgt.

Abbildung 6: Konzept der multiplen Ansichten [59]

Mit diesen Interaktionstechniken für 2D-Eingabegeräte ermöglichen heutige CAD-Systeme sowohl Flächen- als auch Volumenmodellierung.

Einen neuartigen Ansatz, um mit 2D-Eingabegeräten 3D-Objekte zu erzeugen, präsentieren Zeleznik et. al. [184] mit dem System SKETCH. Dabei werden 3D-Primitive aus Linien abgeleitet, die der Benutzer mit der 2D-Maus oder einem Stift in das Graphikfenster skizziert. Methoden der Gestenerkennung dienen dazu, die Linien zu interpretieren, um schließlich ein 3D-Objekt generieren zu können (siehe Abbildung 7). Nach Aussage der Autoren, eignet sich dieser Ansatz in erster Linie zum schnellen und unpräzisen Skizzieren von 3D-Geometrie, daher auch der Name SKETCH. Direktmanipulation wird nur bei der Objektmodifikation, nicht aber bei der Objekterzeugung geboten. Dafür kommt SKETCH ohne Menüs und Buttons aus, allerdings wird für bestimmte Funktionen auf die Tastatur zurückgegriffen. Basierend auf den Ideen von SKETCH wird von Cohen et al. in [38] eine Mensch-Maschine-Schnittstelle zur Eingabe von 3D-Kurven mit 2D-Eingabegeräten beschrieben. Hierbei machen sich die Autoren u.a. die Wechselwirkung zwischen Objekt- und Schattengestalt zunutze, um die räumliche Ausprägung einer 3D-Kurve durch Skizzierung ihres Schattens zu definieren.

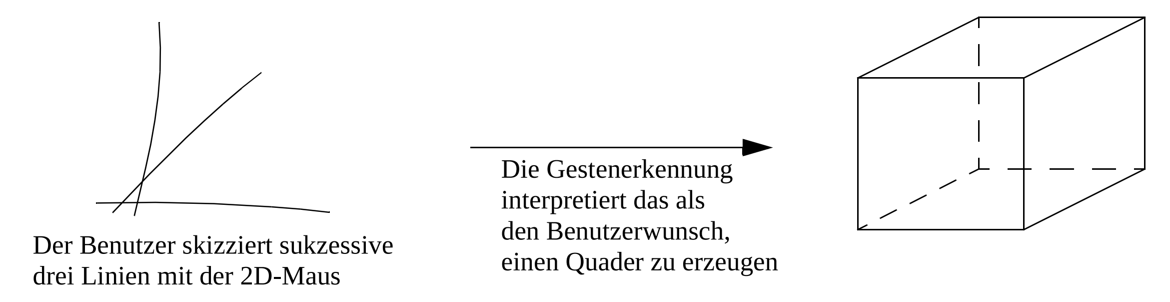


Abbildung 7: Prinzip der Objekterzeugung mit SKETCH am Beispiel eines Quaders

STILTON [165] ist ein anderes System, das die Rekonstruktion drei-dimensionaler Geometrie aus 2D-Gesten in einer orthographischen Sicht erlaubt. Das System kann auch dazu benutzt werden, um 3D-Szenen durch Skizzieren auf einer Fotografie zu erstellen.

Einen innovativen Ansatz zur Freiformflächenmodellierung mit 2D-Eingabegeräten präsentieren Igarashi et. al. [78] in ihrem System Teddy. Aus 2D-Skizzen werden Rotationskörper erzeugt, die topologisch einer Kugel entsprechen. Mit Hilfe von Modifikationsmöglichkeiten lassen sich somit sehr schnell organische Formen modellieren, wie sie in dem Anwendungsbereich Filmindustrie und Unterhaltung oft benötigt werden. Freiformflächen wie sie im Maschinenbau typischerweise vorkommen, können mit diesem Ansatz nicht erzeugt werden. Flexibler bezüglich der erzielbaren Flächenformen ist 'Skin' von Markosian et al. [100]. Dabei handelt es sich um einen Ansatz zum sukzessiven Erzeugen und Verändern polyedrischer Strukturen.

Zheng et al. [186] präsentieren eine Deformationstechnik für Freiformflächen, bei der der Benutzer mit einem 2D-Eingabegerät den zu modifizierenden Bereich auf der Fläche skizzieren und dann manipulieren kann. Andere Forscher setzen zur interaktiven Modellierung *multi-resolution surfaces* [87], Flächen aus irregulären Kurvennetzen [92] und Partikelsystem-basierte Flächen [163] ein.

Zusammenfassend kann festgestellt werden, daß 2D-Eingabegeräte für bestimmte 3D-Interaktionsaufgaben hinreichend gut geeignet sind, so ist z.B. die Technik der virtuellen Kugel intuitiv handhabbar. Auch konnten durch Gestenerkennung Fortschritte auf dem Gebiet des intuitiven, aber unpräzisen Skizzierens erreicht werden. Allerdings bleiben die prinzipiellen Probleme von 2D-Eingabegeräten, wie z.B. mangelnde Reiz-Reaktions-Korrespondenz in 3D, erhalten. Oft ist auch die Direktmanipulation nur eingeschränkt möglich.

2.2.2 3D-Interaktion mit 3D-Eingabegeräten

3D-Interaktion mit 3D-Eingabegeräten wurde vor allem durch die VR-Technologie vorangetrieben. In reinen VR-Anwendungen beschränkt sich 3D-Interaktion oft auf das Greifen und Bewegen von Objekten unter Einbeziehung von Kollisionserkennung. In Modellierungsanwendungen geht es in erster Linie um das Erzeugen und Modifizieren von Geometrie. Im folgenden werden zunächst allgemeine Ansätze zur 3D-Interaktion beleuchtet und dann solche, die sich speziell mit der 3D-Interaktion in Modellierungsanwendungen auseinandersetzen.

2.2.2.1 Allgemeine Techniken

Steed und Slater [148] evaluierten verschiedene Metaphern zur 3D-Interaktion unter Verwendung einer *desktop bat*. Die *desktop bat* ist ein tischgebundenes 3D-Eingabegerät mit 5 Freiheitsgraden. Bestandteil der Untersuchungen war die Navigation sowie die Positionierung von Cursor und Objekten im Raum. Untersuchungen zur Kamerakontrolle (Navigation) mit 3D-Eingabegeräten liegen auch von Ware und Osbourne [175] vor.

Auch Ware und Jessome [174] widmen sich der Frage der Objektplazierung in 3D. Sie benutzen ein fliegendes 3D-Eingabegerät - eine *bat* - mit 6 Freiheitsgraden und wenden verschiedene Manipulationsmodi an, um den Effekt der Visualisierung auf die Positionierung zu untersuchen. Sie kommen zu dem Schluß, daß die *bat* sehr natürliche Interaktionen unterstützt. Ihren Erfahrungen zufolge sind die Ermüdungserscheinungen bei deren Benutzung relativ gering, da der Arm am Ellenbogen aufgestützt werden kann. Ihre Arbeit widmet sich nicht der präzisen Plazierung - auch Ansätze zur Erzeugung graphischer Primitive in 3D sind nicht enthalten.

Felger [57] vergleicht und bewertet verschiedene Eingabegeräte zur Objektpositionierung. Als Eingabegeräte kommen Drehknöpfe, eine Maus, ein SpaceBall und ein Datenhandschuh zum Einsatz. Die zu positionierenden Objekte werden schattiert und perspektivisch dargestellt, die Szene ist von texturierten Wänden umgeben; Schattenwurf kommt nicht zum Einsatz. Der durchzuführende Montageprozeß wird durch *snapping* und Kollisionserkennung unterstützt. Die zusammenzubauenden Objekte können gepickt, rotiert und transliert werden. Als graphisches Echo wird ein *3D full space cursor* verwendet. Die Auswertung ergab, daß die Steuerkugel und der Datenhandschuh der Maus und den Drehknöpfen deutlich überlegen waren - die Aufgabe wurde mit ihnen viel schneller erledigt. Bei erstmaliger Benutzung war der Datenhandschuh wiederum der Steuerkugel überlegen. Das spricht dafür, daß er spontan intuitives Interagieren ermöglicht. Felger stellt schließlich die Vor- und Nachteile der Eingabegeräte gegenüber. Der im Rahmen dieser Arbeit durchgeführte Auswahlprozeß von 3D-Eingabegeräten für das Anwendungsfeld Modellierung zog diese Ergebnisse mit in Betracht.

Eine 3D-Interaktionstoolbox wird von Böhm [24] vorgestellt. Der generische Werkzeugkasten mit Verfahren zur Erkennung von in erster Linie statischen Gesten bietet 3D-Interaktionstechniken vor allem für VR. In beispielhaften Anwendungen wird u.a. die Manipulation von triangulierter Geometrie mittels Selektion und Modifikation von Eckpunkten sowie das Skalieren von Objekten durch Greifen von Handles gezeigt. Die multi-modale Eingabe stellte einen weiteren Schwerpunkt der Arbeit dar. Das Anwendungsfeld CAD lag hingegen nicht in deren Fokus.

Bimber führt in [20] einen Ansatz zur Erkennung dynamischer Gesten zur Mensch-Maschine-Interaktion in virtuellen Umgebungen ein. Er setzt dazu Verfahren aus der Künstlichen Intelligenz und Neuroinformatik sowie dem Compilerbau zur Erkennung hierarchisch aufgebauter Gesten ein. Durch Kombination dieser Verfahren läßt sich die Gestenerkennung für verschiedene Anwendungen konfigurieren: zur Erkennung von Schriftzeichen genauso wie zur Interpretation von 2D- und 3D-Gesten. Die 2D- und 3D-Gestenerkennung kann zum unpräzisen Skizzieren von Geometrieobjekten eingesetzt werden [54].

2.2.2.2 Techniken zur Modellierung von 3D-Objekten

Mit 3DM wurde von Butterworth et al. [30] erstmals ein immersives Modellierungssystem vorgestellt, das für die Benutzung eines Head-Mounted-Displays ausgelegt ist. Der Benutzer kann in der virtuellen Welt navigieren und Objekte aus Standard-Primitiven, wie Kugeln oder Zylinder, zusammensetzen. Für die Modellierung von Flächen werden Dreiecksnetze verwendet. Echte Freiformflächenmodellierung wird nicht unterstützt. Einen ähnlichen Ansatz verfolgt CHIMP [101], bei dem der Benutzer ein- oder gar zweihändig mit Datenhandschuhen in VR modellieren kann. CHIMP wurde zu einer verteilten VR-Umgebung, in der auch modelliert werden kann, weiterentwickelt [39].

Dani und Gadh beschreiben in [44] und [36] ihre Entwicklungen rund um das System COVIRDS (Conceptual Virtual Design System) - ein Modellierungssystem für den konzeptionellen Entwurf. Ausgabeseitig setzen die Autoren semi-immersive Technologie ein, wie stereoskopische Großbildprojektion bzw. Virtueller Tisch. Eingabeseitig läßt das System multisensorische Daten zu, so können Informationen von bis zu zwei Datenhandschuhen und Spracheingaben verarbeitet werden, um Objekte zu erzeugen und deren Parameter zu spezifizieren. Trotz 3D-Eingabetechnologie greifen die Autoren auf indirekte Interaktionen mittels eines Pickstrahls zurück. Die in [45] beschriebenen Systemfunktionalitäten zur Freiformflächenmodellierung beschränken sich auf die Selektion und Modifikation von Kontrollpunkten. Ausgangspunkt ist dabei eine ebene Fläche mit wählbarer Anzahl von Kontrollpunkten. Die Erzeugung neuer Freiformflächen im Raum wird

nicht unterstützt. Ansätze, die der reinen Manipulation von Kontrollpunkten in bezug auf Intuitivität und Voraussagbarkeit des Ergebnisses überlegen sind, sind als *free from deformation* [135] und *extended free-form deformation* [41] bekannt.

Sachs, Roberts und Stoops [126] präsentieren mit 3-Draw ein System, mit dem der Benutzer mittels zweier 3D-Eingabegeräte Raumkurven und -netze skizzieren kann. In der einen Hand hält er ein frei bewegliches Tablett und in der anderen einen Stift - beide Eingabegeräte sind positionsgetrackt. Führt der Benutzer nun den Stift durch den Raum, entsteht eine 3D-Kurve; die Kurven können editiert werden. Durch Drehen des Tabletts, kann die Orientierung des entsthenden Drahtgittermodells geändert werden. Als Ausgabegerät kommt ein gewöhnlicher Monitor zum Einsatz. Zur Verstärkung des räumlichen Eindrucks wird ein virtueller Boden eingesetzt, auf den das Drahtmodell projiziert wird. Um das Monitorbild nicht zu verdecken, muß neben dem Monitor gearbeitet werden, was die Augen-Hand-Synchronisierung erschwert. Sachs et al. kommen zu dem Schluß, daß mit ihrem System in den frühen Phasen der Konstruktion, insbesondere im Design (styling and shaping), ein schnelleres Skizzieren und direkteres Arbeiten in 3D möglich ist als mit herkömmlichen Verfahren, wenn sich der Benutzer erst einmal an das neue Paradigma gewöhnt hat. Ein Anwendungsbeispiel für die Generierung von Kurven in 3D an einem tischähnlichen Rückprojektionssystem (ErgoDesk) wird von Forsberg et al. in [64] beschrieben. Die Problematik der Augen-Hand-Koordination entfällt hierbei, da Kurven mittels direkter Interaktion in 3D bei stereoskopischer Darstellung erzeugt werden können.

Hummels et al. untersuchen in [76] und [77] die Arbeitsmethode von Autodesignern und schlagen eine virtuelle Umgebung von, die gesten-basiert Unterstützung für das Styling geben soll. Leider werden keine Implementierungshinweise gegeben bzw. Resultate dargelegt. Eine Untersuchung desselben Anwendungsfeldes liefern Fontana et al. [60]; sie leiten daraus einen Bedarf für Features in der Freiformflächengestaltung ab.

Einen innovativen Ansatz zum schnellen Erzeugen flächiger Geometrie beschreiben Schkolne und Schröder in [130]. Hierbei werden an einem Virtuellen Tisch durch Gesten mit einem Datenhandschuh Dreiecksnetze erzeugt, die flächenhaft dargestellt werden. Durch schnelle Re-Triangulierung können Teilflächen eingefügt und somit bereits existierende Flächen interaktiv modifiziert werden. Das System ermöglicht den experimentellen Entwurf von Flächen, erzeugt aber keine Freiformflächen im strengen Sinne. Zur Generierung von z.B. NURBS [114] müsste das Dreiecksgitter einem Interpolations- bzw. Approximationsverfahren unterzogen werden. Anforderungen nach Symmetrie wie sie beim Entwurf industrieller Produkte oft anzutreffen sind, trägt der Ansatz nicht Rechnung.

Die virtuelle Tonbearbeitung [97] versucht den klassischen Modellbau nachzuempfinden und stellt im weiteren Sinne ebenfalls einen Ansatz zur Modellierung von Freiform-Geometrie dar. Ausgehend von einem Voxelmodell werden durch Bewegen von virtuellen Schablonen durch den Voxelraum subtraktive bzw. additive Operationen ausgeübt. In jedem Interaktionsschritt erfolgt eine Triangulierung des Voxelmodells mit Hilfe des Marching-Cube-Algorithmus. Diese Umsetzung der interessanten Idee, den traditionellen Modellbau mit Ton zu virtualisieren, ist mit zwei Nachteilen behaftet. Erstens hemmt der berechnungsaufwendige Marching-Cube-Algorithmus die Interaktivität der Umsetzung, zweitens führt die gewählte Datenstruktur - das Voxelmodell - zu unschönen Treppeneffekten, die durch den Marching-Cubes zwar kompensiert aber nicht aufgehoen werden. Auch dieser Ansatz generiert also keine Freiformflächen im engen Sinne des Begriffes. Weitere Ansätze zur Modellierung beliebig geformter Volumenkörper werden in [67] und [105] als 'sculpting' präsentiert.

Liang und Green präsentierten mit JDCAD [94] ein 3D-Modellierungssystem, welches ein hohes Maß an Interaktivität bietet. Der Benutzer arbeitet an einem normalen CAD-Arbeitsplatz mit einer *bat* - einem fliegenden 3D-Eingabegerät mit 6 Freiheitsgraden - und einem Head-Tracker. Der Head-Tracker wird benutzt, um ein kinetisches Display zu realisieren, d.h. Kopfbewegungen werden ähnlich wie in VR auf Bewegungen der Kamera übertragen, somit wird ein Tiefeneffekt durch Bewegungsparallaxe erzielt. Die *bat* steuert einen 3D-Cursor, der als kleines Koordinatenkreuz im Raum ausgeprägt ist. Die Selektion geschieht über einen Strahl, der vom Cursor ausgesendet und im Raum orientiert werden kann. Das System erlaubt die Erzeugung von Objekten mit dem 3D-Cursor, dabei wird zunächst der Objekttyp selektiert und dann mit dem Cursor eine Bounding-Box aufgezogen, die die Größe des Objektes bestimmt; dabei werden die Höhe, Breite und Tiefe der Bounding-Box auf drei Objektparameter abgebildet. Hat ein Objekt mehr als drei Parameter, werden die anderen Parameter relativ zu den aus der Bounding-Box abgeleiteten bestimmt. Eine Bounding-Box-basierte Methode zur Objekterzeugung hat den Nachteil, daß sich der Objektmittelpunkt während der Erzeugung verschiebt (siehe Abbildung 8). Dies ist gerade in der Konstruktion unerwünscht, wo Objekt- bzw. Flächenmittelpunkte oft Bezugs- oder Fixpunkte darstellen. Alle Parameter können nachträglich mit der *region-based reshaping*-Methode geändert werden. Problematisch bei dieser Methode sind Überlappungen der sensitiven Regionen, die mit zunehmender Komplexität der Modelle immer wahrscheinlicher werden. Präzise Parametereinstellungen können mit dem *bat-operated dial* vorgenommen werden, einem Wertgeber, der mit dem 3D-Eingabegerät gesteuert wird. Vergleiche mit kommerziellen Systemen zeigen - so Liang und Green -, daß das Kreieren eines 3D-Modells nur ein Zehntel bis ein Fünftel der Zeit in Anspruch nimmt; die Bedingungen, unter denen der Vergleichstest stattfand, sind nicht näher beschrieben. Die Aussage zeigt aber, welches Potential in der 3D-Eingabe zumindest für das Skizzieren 3-dimensionaler Modelle steckt.

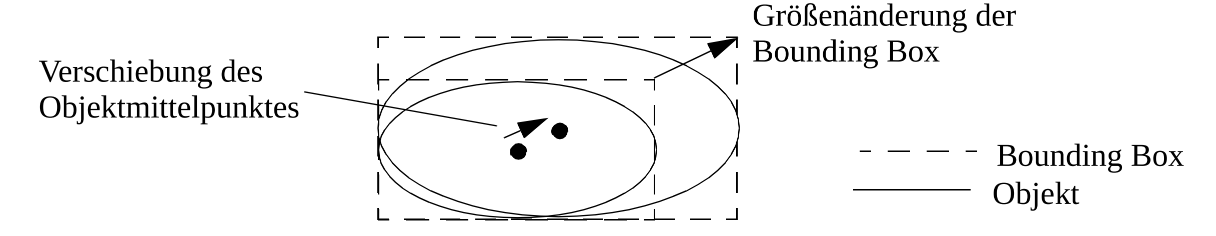


Abbildung 8: Aus BoundingBoxen abgeleitete Objekterzeugung im 2-dimensionalen Fall

VADE (Virtual Assembly Design Environment) von Jayaram et. al. [84] ist eine immersive Umgebung zur Zusammenbausimulation mit einer Schnittstelle zu Pro/Engineer, die es erlaubt, die in Pro/Engineer generierten Constraints für die Zusammenbausimulation heranzuziehen. VADE bietet keine Modellierfunktionalität und ist daher eher als konventionelles VR-System zu bezeichnen, das zur Montage-/Demontagesimulation eingesetzt wird. Seine Besonderheit ist die enge Ankopplung an Pro/E und die Verwendung von Constraints aus Pro/E in der Zusammenbausimulation.

Die Mehrzahl der sog. VR-CAD-Systeme setzt eingabeseitig auf Datenhandschuhe. Die Akzeptanzprobleme dieser Eingabegeräte überschatten das Potential von 3D-Eingabe für Modellierungsaufgaben. Endanwender, die sich von der Gerätetechnologie schockiert erst einmal abgewendet haben, werden die software- und interaktionsseitigen Vorteile und Möglichkeiten nicht wahrnehmen. Daher ist es wichtig, Wege zum ad hoc Einsatz von VR-Technologie in Modellierungsanwendungen zu finden.

2.2.3 Zweihändige Interaktionstechniken

In unserem täglichen Leben verwenden wir oft beide Hände, um eine Aktion auszuführen und sind in bestimmten Situationen sogar hilflos, wenn wir nur eine Hand zur Verfügung haben. Das Arbeiten mit zwei Händen ermöglicht die Parallelisierung von Tätigkeiten, die normalerweise nacheinander ausgeführt werden müssen. Dies wurde auch von Forschern auf dem Gebiet der Mensch-Maschine-Interaktion erkannt und schlägt sich in Experimenten und der Entwicklung zweihändiger Interaktionstechniken nieder.

Schon 1986 wurden von Buxton und Myers zwei Experimente zur zweihändigen Eingabe durchgeführt [31]. Eine Gruppe von Testpersonen führte die Tests einhändig durch, die andere zweihändig. Die 'zweihändigen' Testpersonen agierten im Durchschnitt über 40% der Zeit parallel und benötigten zur Durchführung ihrer Aufgabe weniger Zeit. Auch Leganchuk, Zhai und Buxton [93] bestätigen in zwei Versuchen, daß die zweihändige Eingabe Geschwindigkeitsvorteile gegenüber der einhändigen Eingabe bietet. In den durchgeführten Experimenten zeigte die zweihändige Eingabe einen Zeitvorteil von 17% bzw. 35% gegenüber der einhändigen Eingabe.

Forsberg, Strauss und Zeleznik verwenden in einer Weiterentwicklung von SKETCH zwei 2D-Mäuse und ein Grafiktablett, das den simultanen Gebrauch eines Drei-Tasten-Stiftes und einer Vier-Tasten-Maus (4-button puck) zuläßt [63]. Sie haben Techniken entwickelt, um Objekte durch die Verwendung beider Hände z.B. gleichzeitig zu rotieren und zu translieren. Ebenso haben sie Techniken zum Skalieren von Objekten, zur Kamerakontrolle und zur Erzeugung von graphischen Primitiven entwickelt, die einfacher und intuitiver zu bedienen sind als herkömmliche Techniken mit einer 2D-Maus. Interessant ist auch das Interagieren mit den Schatten, so können z.B. Objekte in der Ebene des Schattens oder orthogonal dazu verschoben werden.

Shaw und Green [140] präsentieren mit THRED (*für Two Handed Refining EDitor*) einen Ansatz zum zweihändigen Flächendesign, wobei sie sich auf polygonale Oberflächen beschränken. Beiden Händen kommt eine unterschiedliche Bedeutung zu. Die linke Hand setzt den Constraint-Modus und orientiert die gesamte Szene im Raum, die rechte ist für die Selektion von Kontrollpunkten und deren Veränderung verantwortlich. Als Eingabegeräte werden selbst entwickelte "button-enhanced bats" verwendet. Die bats bestehen aus einem Polhemus-Sensor und drei Druckschaltern zur Befehlseingabe und sind sehr klein und leicht gehalten.

Turner, Gobetti und Soboroff stellen in [164] ein Verfahren zum zweihändigen Kreieren von animierten Charakteren vor. Sie verwenden einen Headtracker und eine Shutterbrille, um einen besseren 3D Eindruck von der generierten Szene zu vermitteln. Als Eingabegeräte dienen ein Spaceball für die linke Hand und eine 3D-Ultraschall-Maus für die rechte Hand. Die linke Hand wird verwendet, um durch die Szene zu navigieren, und um Objekte zu bewegen. Mit der rechten Hand werden die Objekte manipuliert.

Mit zwei Datenhandschuhen und einem Stift (mit je einem Polhemus-Sensor) arbeiten Cutler, Fröhlich und Hanrahan an der Responsive Workbench [43]. Sie verwenden eine Toolbox mit 3D-Icons, um verschiedene Interaktionen auszulösen sowie Gesten, um z.B. die Freiheitsgrade der Eingabe mit der dominanten Hand einzuschränken. Obwohl beiden Eingabegeräten dieselbe Trackingtechnologie zugrundeliegt, wird nach Aussage der Autoren für präziseres Arbeiten der Stift bevorzugt, während grobes Arbeiten mit den Datenhandschuhen möglich ist. Die verschiedenen Tools werden unterteilt in Einhandtools, symmetrische Zweihandtools und asymmetrische Zweihandtools. In Experimenten sind Versuchspersonen sehr schnell dazu übergegangen, zwei unterschiedliche Einhandtools zu verwenden.

Einen anderen Weg gehen Bier und Stone mit ihren *Toolglasses* und *Magic Lenses* [18], [19]. Dies sind durchsichtige Werkzeuge, die zwischen dem Mauszeiger und der Applikation angeordnet sind. Sie können mit der nicht-dominanten Hand bewegt werden. Um z.B. die Farbe einer Fläche zu ändern, fährt man mit der transparenten Palette über die zu füllende Fläche bis der gewünschte Farbton über der Fläche zu liegen kommt und klickt mit der Maus 'durch die Palette'. Außerdem ist es möglich, das Werkzeug so zu gestalten, daß es die darunterliegenden graphischen Objekte vergrößert, als Linienmodell darstellt oder eine Vorschau liefert. Dies sind nur einige Beispiele für diese Werkzeuge, die nahezu beliebige Funktionen bereitstellen können.

2.3 3D-Ausgabegeräte

3D-Ausgabegeräte wurden entwickelt, um dem Benutzer computer-generierte Szenen 3-dimensional präsentieren zu können. Durch eine 3-dimensionale Darstellung wird das Erkennen räumlicher Beziehungen, das für ein zielgerichtetes Interagieren notwendig ist, positiv beeinflußt. Im folgenden werden verschiedene Geräte vorgestellt, die die stereoskopische Darstellung unterstützen. Zunächst erfolgt allerdings eine Zusammenstellung der Anforderungen an diese Geräte aus der Sicht der Modellierung. Abschließend werden die vorgestellten Geräte hinsichtlich dieser Anforderungen bewertet.

2.3.1 Anforderungen an 3D-Ausgabegeräte aus der Sicht von CAD

An CAD-Arbeitsplätze werden gerade bezüglich der Ausgabe spezielle Anforderungen hinsichtlich der Ergonomie gestellt, die in diversen Richtlinien, z.B. der EU-Bildschirmarbeitsplatzrichtlinie, festgeschrieben sind. Heutige Gerätetechnik zur 3D-Ausgabe kann die dort gestellten Anforderungen kaum erfüllen, so daß es noch geraume Zeit dauern wird, bis ergonomische 3D-Ausgabe möglich sein wird. Die 3D-Ausgabe sollte folgende Forderungen erfüllen:

- Eine möglichst hohe Bildqualität und Bildwiederholfrequenz bieten.
- Ermüdungsfrei und angenehm zu benutzen sein.
- Den Blick auf den Arbeitsplatz erlauben.
- In Relation zum CAD-Arbeitsplatz preiswert sein, sonst wird sie vom Markt nicht angenommen.

2.3.2 Der Monitor

Der Monitor ist das klassische Ausgabegerät im CAD. Moderne Monitore und entsprechende Graphik-Hardware ermöglichen die stereoskopische Darstellung von CAD-Modellen entweder in aktiver oder - mittels eines Filters - in passiver Stereotechnik. Bei der aktiven Stereoprojektion trägt der Benutzer eine aktive Stereobrille (siehe Abbildung 9). Auf dem Monitor wird im schnellen Wechsel ein Bild für das rechte bzw. das linke Auge erzeugt. Die aktive Brille sorgt dafür, daß zu jedem Zeitpunkt das jeweilige Bild von dem richtigen Auge wahrgenommen wird, in dem die Brillengläser alternierend verdunkelt werden. Durch hinreichend schnellen Wechsel der Bilder (~ 60 Hz) sind die Augen aufgrund ihrer Trägheit nicht in der Lage, die beiden disparaten Bilder aufzulösen und es entsteht ein 3-dimensionaler Eindruck von der betrachteten virtuellen Szene. Bei der passiven Technik sitzt vor dem Monitor ein Polarisationsfilter, das mit jedem Bildwechsel umgeschaltet wird. Der Benutzer trägt eine leichte Polfilterbrille, die für das jeweilige Auge nur das Licht durchläßt, welches entsprechend polarisiert ist.

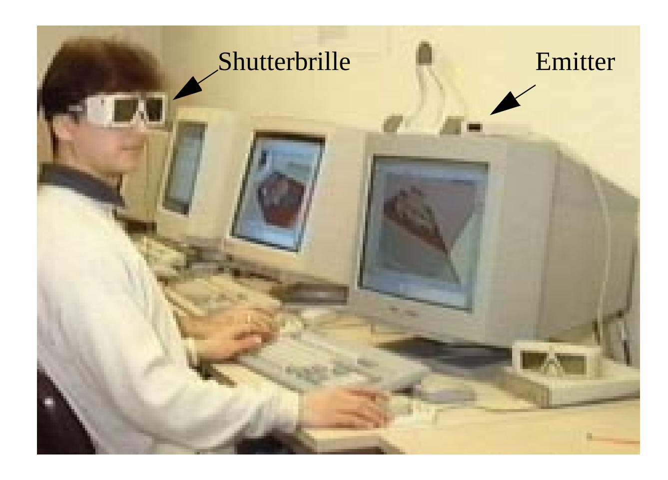


Abbildung 9: Benutzer mit aktiver Shutterbrille am CAD-Arbeitsplatz

Bei der Stereo-Ausgabe verdoppeln sich die Anforderungen an die Graphikleistung der Hardware bzw. es halbiert sich die Wiederholrate bei gleicher Graphikleistung. Des weiteren wird für eine Stereo-Darstellung mit voller Bildschirmauflösung doppelt soviel Bildschirmpuffer (frame buffer) benötigt wie in der monoskopischen Darstellung. Steht nicht genügend Bildspeicher zur Verfügung, so wird im Stereomodus die vertikale Auflösung halbiert. Aus diesen Gründen waren CAD-Workstations in der Vergangenheit kaum in der Lage, stereoskopische Darstellungen mit angemessenen Bildwiederholraten und Auflösungen zu generieren. Dies hat sich erst in den letzten Jahren geändert. Seit kurzer Zeit ist die stereoskopische Ausgabe auch auf PCs möglich.

2.3.3 Der Virtuelle Tisch

Der Virtuelle Tisch[1] ist ein stereoskopisches Rückprojektionssystem, vergleichbar mit einem großen, flachliegenden Monitor in Form eines Tisches (siehe Abbildung 10). Mit Hilfe aktiver Shutterbrillen ist die stereoskopische Ausgabe möglich, dabei dient ein 3-Strahl-Röhrengerät als Projektor. Aufgrund seiner Größe und Eigenschaften bietet der Virtuelle Tisch neue Formen der Präsentation und Interaktion. Anders als traditionelle VR-Gerätetechnologie, wie Datenhelm und Datenhandschuh, läßt er sich leichter in gewohnte Arbeitsumgebungen integrieren. Er verbindet Eigenschaften des traditionellen Zeichenbretts mit solchen virtueller Umgebungen und erscheint daher auch für den Modellierungsprozeß als geeignet [53].

Die Einwicklung geeigneter Interaktionstechniken ist die eigentliche Herausforderung, die der Virtuelle Tisch stellt. Bei stereoskopischer Ausgabe scheint das 3D-Modell über dem Tisch zu schweben, so daß es naheliegend ist, Interaktionen direkt im Raum über dem Tisch ausführen zu wollen. Der im Rahmen dieser Arbeit verwendete Virtuelle Tisch zeichnet sich durch folgende Merkmale aus:

- er ist kompakt,
- er bietet eine hohe Bildqualität in bezug auf Schärfe und Kontrast und
- er ist in der Neigung verstellbar, ohne daß die Projektionseinheit nachjustiert werden muß.

1. Das Prinzip des Virtuellen Tisches geht auf die Responsive Workbench [91], eine Entwicklung der GMD, zurück.

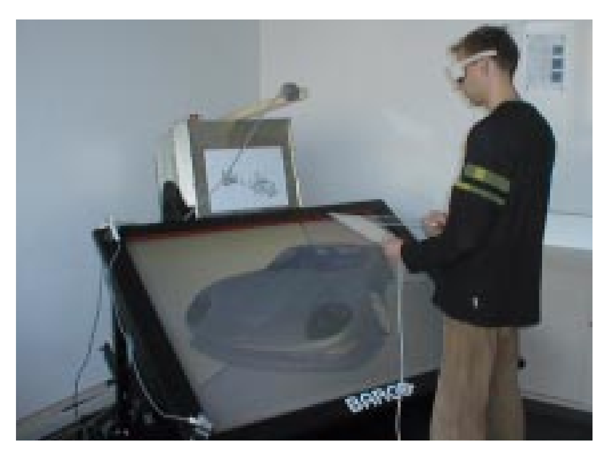


Abbildung 10: Virtueller Tisch [14] - Design Review mit ARCADE

2.3.4 Das Head Mounted Display

Ein Head Mounted Display (HMD) - auch Datenhelm genannt - wird vom Benutzer auf dem Kopf getragen. Zwei Bilderzeuger sorgen dafür, daß jedem Auge ein entsprechendes Bild zugeführt wird und so ein 3D-Eindruck von der Szene entsteht. Datenhelme werden üblicherweise mit einem Head-Tracker kombiniert, der die aktuelle Position und Orientierung des Kopfes liefert. Die Kopfposition wird ausgewertet, um dem Benutzer den jeweiligen Ausschnitt der virtuellen Welt im Datenhelm zu präsentieren. So ist der Benutzer stets in die virtuelle Welt integriert, egal wohin er schaut. Dies ist ein großer Vorteil gegenüber den meisten stationären Projektionssysteme, wie z.B. dem Virtuellen Tisch oder Projektionswänden, allerdings wird dieser Vorteil mit den Nachteilen erkauft, daß das Bild 'nachläuft' und HMDs bei guter Bildqualität schwer sind und auf Dauer wenig Tragekomfort bieten.

Neben den klassischen Datenhelmen existieren auch halb-transparente HMDs, die für Anwendungen der erweiterten Realität (Augmented Realitiy - AR) eingesetzt werden. Dabei wird die reale Szene, die durch das halb-transparente Display weiterhin sichtbar bleibt, mit computergenerierter Information angereichert. Anwendungen von AR sind heute vor allem im Bereich Training und Wartung zu finden [13].

2.3.5 Sonstige 3D-Ausgabegeräte

Neben den hier vorgestellten gibt es eine Menge andere Ausgabegeräte, die kurz erwähnt werden sollen. Das BOOM ist ein bilderzeugendes System, in das der Benutzer ähnlich wie in ein Mikroskop hineinschaut. Es ist variabel aufgehängt. Die Bewegungen des BOOMs werden auf die virtuelle Kamera übertragen, wodurch die computer-generierte Szene entsprechend neu dargestellt wird.

Die CAVE [42] ist ein Raum, auf dessen Wänden durch Rückprojektion virtuelle Welten dargestellt werden können. Die CAVE eröffnet eine neue Dimension von Immersion: Erstmals ist es möglich, daß der Betrachter seinen eigenen Körper in der virtuellen Umgebung wahrnimmt. Dies verhilft zu einer neuen Dimension der Präsenz innerhalb virtueller Umgebungen.

Darüber hinaus gibt es Entwicklungen, die die Bildinformation direkt auf die Netzhaut schreiben (retina displays), wobei ein verträglicher Laser eingesetzt wird. Auch die Darstellungsqualität auto-stereoskopischer Displays - also von 3D-Ausgabegeräten, die seitens des Betrachters ohne Stereobrille auskommen - wird stetig verbessert.

2.3.6 Bewertung der 3D-Ausgabegeräte aus Sicht der CAD-Anforderungen

Stereobrillen bieten in Verbindung mit hochauflösenden Monitoren eine gute Bildqualität und hinreichend hohen Kontrast. Sie sind relativ komfortabel zu tragen, wobei eine passive Brille leichter als eine aktive ist. In Relation zu einem HMD mit gleicher Auflösung ist diese Technologie sehr preiswert. Außerdem erlauben die Brillen dem Benutzer den Blick auf den Arbeitplatz. Aktive Brillen werden durchsichtig, wenn der Betrachter seinen Kopf vom Bildschirm abwendet, um z.B. ein auf dem Schreibtisch liegendes Dokument zu lesen; passive Brillen sind ohnehin durchsichtig.

Head-Mounted Displays weisen folgende Nachteile auf, die sie für den dauerhaften Einsatz am Arbeitsplatz ungeeignet erscheinen lassen:

- Das Tragen ist unkomfortabel.
- HMDs ohne see-through-Fähigkeiten müssen abgesetzt werden, möchte der Benutzer die reale Umwelt wahrnehmen.
- Die Kopplung der Szene an Kopfbewegungen ist realitätsnah, ruft jedoch oft Unbehagen hervor, weil die Wahrnehmung der virtuellen Welt nicht mit der Information, die vom Gleichgewichtssinn erzeugt wird, übereinstimmt.
- Helme, die hochwertige Bilder (hohe Auflösung und guten Kontrast) liefern, sind relativ schwer und teuer.

Zusammenfassend läßt sich festhalten, daß Stereobrillen in Kombination mit dem Monitor oder Projektionssystemen für den CAD-Einsatz geeigneter erscheinen als Darstellungshelme.

2.4 Die Rolle der Visualisierung für Wahrnehmung und Interaktion

Soll die Effizienz im Umgang mit 3D-CAD gesteigert werden, stellt sich nicht nur die Frage nach den Interaktionstechniken, sondern auch danach, wie durch geeignete Visualisierung der effizienten Interaktion beigetragen werden kann. Die Basis zur Entwicklung geeigneter Visualisierungstechniken ist die Beachtung der wahrnehmungspsychologischen Prinzipien des menschlichen Sehens. Eine essentielle Aufgabe, um die Orientierung, Navigation und Positionierung in 3D CAD-Systemen zu verbessern, ist es, die Objekte so darzustellen, daß der Betrachter einen aufschlußreichen Form- und Tiefeneindruck bekommt. Eine weitere wichtige Aufgabe der Visualisierung ist die schnelle Rückkopplung der Interaktionen an den Benutzer.

2.4.1 Das menschliche Sehen

Der Mensch nimmt seine Umwelt vor allem visuell wahr. Das Sehen ist der am stärksten ausgeprägte und wichtigste Sinn; über 80% aller Sinneswahrnehmungen erfolgen über die Augen. Vom visuellen System werden eine Vielzahl von Informationen ausgewertet, um aus den von den beiden Augen wahrgenommen Bildern eine 3-dimensionale Gestalt zu rekonstruieren. Bei 2-äugigem (binokularem) Sehen spielen Disparität und (Kon-)Vergenz[2] eine entscheidende Rolle bei der Tiefenwahrnehmung [123]. Unter Konvergenz versteht man die Anwinkelung der Augen zum betrachteten Objekt. Disparität bezeichnet die durch den unterschiedlichen Betrachtungswinkel entstehenden Differenzen in den von den beiden Augen wahrgenommenen Bildern (siehe Abbildung 12). Im strengen Sinne erfolgt die Wahrnehmung erst im visuellen System und Gehirn

2. In der Literatur werden die Begriffe Vergenz und Konvergenz synonym verwendet.

des Menschen, das Auge ist ’nur’ das bildgebende System, wobei erste Bildauswertungen schon vom Sehnerven durchgeführt werden.

Neben der Disparität und der Konvergenz gibt es Form- und Tiefenhinweise, die sowohl beim 2-äugigen als auch beim 1-äugigen Sehen - dort zwangsweise - vom Wahrnehmungsapparat zur Bildinterpretation herangezogen werden. Diese werden im folgenden näher dargestellt.

2.4.1.1 Monokulare Form- und Tiefenhinweise

Auch wenn der Mensch gewöhnlich zweiäugig sieht, ist er in der Lage, aus monokular wahrgenommenen Szenen, Form- und Tiefeninformation zu extrahieren. Die binokularen Merkmale treten bei zunehmender Betrachtungsentfernung in den Hintergrund, da die Augen dann praktisch parallel ausgerichtet sind.

Die Form- und Tiefenwahrnehmung basiert bei monukuralem Sehen auf folgenden Effekten:

- **Größenunterschiede** der betrachteten Objekte lassen auf die Entfernung schließen, sofern die absolute Größe der Objekte bekannt ist (siehe Abbildung 11a).
- **Verdeckungen** zeigen dem Betrachter, welches Objekt vor welchem anderen liegt (siehe Abbildung 11b).
- **Perspektivische Verzerrungen** geben Hinweise auf Form und Abstand des Objektes vom Fluchtpunkt (siehe Abbildung 11g).
- **Schattierung** ist ein wichtiger Informationsträger für die Form eines Objektes, allerdings läßt sich der Mensch leicht täuschen, was Konkavitäten und Konvexitäten anbelangt, da die gewohnte Lichtquelle - die Sonne - die Umgebung von oben beleuchtet.
- **Schattenwurf**: Schatten können sowohl Form als auch Tiefeninformation enthalten. Aus ihnen läßt sich leicht auf die relative Position der Objekte schließen. Auch auf die Form des schattenwerfenden Objektes lassen sich Rückschlüsse aus der Kontur des Schattens ziehen (siehe Abbildung 11h).
- **Helligkeit**: zwei Objekte mit identischen Oberflächen erscheinen bei unterschiedlichem Abstand von der Lichtquelle verschieden hell; dies ist ein weiterer Anhaltspunkt, um Tiefe abzuschätzen. Besitzen die Objekte unterschiedlich helle oder reflektierende Oberflächen, entfällt dieses Merkmal und kann irreführen.
- **Atmosphärische Effekte**: Unsere Atmosphäre dämpft Licht um ein gewisses Maß, um so mehr, je dichter sie ist. In natürlichen Szenen mit großer Tiefe werden aus atmosphärischen Effekten, wie z.B. Dunst und Nebel, Positionsinformationen abgeleitet (siehe Abbildung 11f).
- **Textur und Texturkompression** geben sowohl Form- als auch Tiefenhinweise. Formhinweise, da die Textur durch die Oberflächengestalt des Objektes moduliert ist. Tiefenhinweise, weil Texturelemente bei zunehmender Entfernung durch die Perspektive immer kleiner werden und näher zusammenrücken (siehe Abbildung 11e).
- **Akkomodation**, also die Brennweiteneinstellung der Augen, festgestellt durch die aufgebrachte Muskelspannung an den Sehmuskeln, wird bei kleinen Betrachtungsabständen benutzt, um diese abzuschätzen. Die Akkomodation läßt bei geringen Betrachtungsabständen eine sehr genaue Entfernungseinschätzung zu (siehe Abbildung 11c).
- **Relative Geschwindigkeit** (Bewegungsparallaxe) in bewegten Szenen gibt Auskunft über die Tiefenebene in der sich bestimmte Objekte befinden, da sich näher am Betrachter befindliche Objekte schneller bewegen als weiter entfernt liegende (siehe Abbildung 11d).

- **Tiefenunschärfe**: Fokussiert ein Betrachter einer realen Szene auf einen bestimmten Betrachtungsabstand, so werden Objekte, die vor und hinter dieser Ebene liegen, unscharf. Dies ist ein Indiz dafür, daß sie eine andere Entfernung zum Betrachter besitzen; besonders auffällig ist dieser Effekt häufig bei Fotografien.

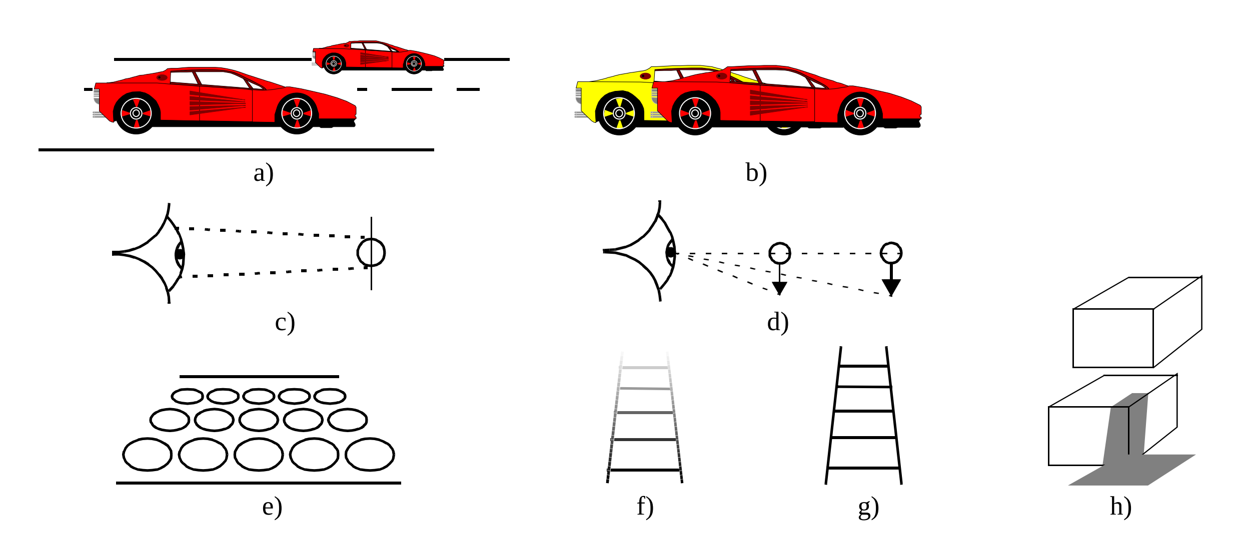


Abbildung 11: Monokulare Tiefenhinweise

2.4.1.2 Binokulare Tiefenhinweise

Wie bereits erwähnt, ergänzen beim binokularen Sehen die Disparität der wahrgenommenen Bilder - hervorgerufen durch die Konvergenz der Sehachsen - bei geringen Betrachtungsabständen den Eindruck von Tiefe. Experimente mit Zufallsstereogrammen haben gezeigt, daß Form- und Tiefenwahrnehmung unabhängig voneinander funktionieren [85], so daß sich schlußfolgern läßt, daß die monokularen Tiefenmerkmale abgeleitete Merkmale sind und nur stereoskopisches Sehen inhärente Tiefenmerkmale bietet. Ein Tiefeneindruck läßt sich also aus monokularen Bildern ableiten, kommt jedoch einem binokularen Sehen nicht gleich.

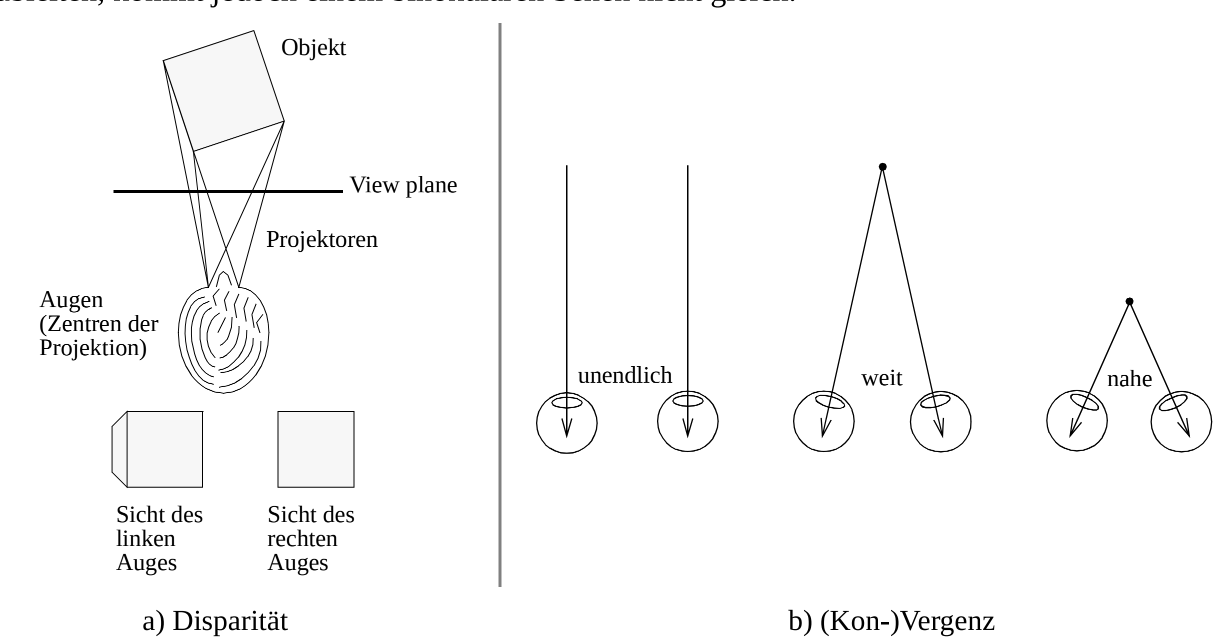


Abbildung 12: Binokulares Sehen [123]

2.4.2 Untersuchungen zur Bedeutung der Visualisierung in der Computergraphik

In diesem Abschnitt werden einige wichtige Arbeiten vorgestellt, die sich mit der Rolle der Visualisierung, insbesondere im Hinblick auf Interaktion, auseinandersetzen.

Forrest [61] diskutiert die Rolle der Benutzungsoberflächen für die 3-dimensionale geometrische Modellierung kritisch und adressiert dabei insbesondere die Bedeutung der Visualisierung. Seiner Meinung nach gibt es zwei unterschiedliche Anforderungen: Erstens die rasche Erfassung der 3-dimensionalen Gestalt und der räumlichen Beziehungen von Objekten und zweitens die präzise Generierung und Positionierung von Objekten. Er stellt fest, daß es viele Systeme gibt, die eine Anforderung erfüllen, aber nicht beide gleichzeitig. Bezüglich der Visualisierung und Interaktion kommt er zu dem Schluß, daß es in der Konstruktion nicht eine optimale Darstellungsform gibt, sondern daß dem Benutzer die Möglichkeit gegeben werden sollte zu wählen und zu kombinieren, z.B. schattierte Darstellung überlagert mit Drahtgitter-Darstellung. Solche Kombinationen werden von modernen CAD-Systemen mittlerweile angeboten. Für die Interaktion stellt er die Rolle von Einschränkungen (constraints) dar und empfiehlt, den 3D-Konstruktionsprozeß möglichst weitgehend auf 2D abzubilden, da das Arbeiten in 3D inhärent schwieriger sei. Er kann seine Aussagen allerdings nicht untermauern und weist auf die Notwendigkeit von Evaluierungen hin, um den Wert verschiedener Visualisierungs- und Interaktionstechniken zu ermitteln.

Kjelldahl und Prime [86] legen eine Studie vor, die untersucht, wie die Tiefenwahrnehmung 3-dimensionaler Objekte auf einem 2D-Bildschirm von der Darstellung im Drahtgitter bzw. im schattierten Modus beeinflußt wird. Die Untersuchungen resultieren in der Feststellung, daß die Darstellungsart kaum Einfluß auf die Tiefenwahrnehmung hat. Vielmehr entscheidet die Anordnung der Objekte über die Genauigkeit der Abschätzung seitens der Testpersonen. Als zukünftige Arbeiten haben sich die Autoren vorgenommen, den Einfluß von Schattenwurf, z.B. auf einen imaginären Boden, und die Wirkung stereoskopischer Ausgabe zu untersuchen.

Wanger et al. [173] führten weitergehende Analysen durch. Dabei wurden verschiedene visuelle Hinweise für die Wahrnehmung räumlicher Relationen in computer-generierten Bildern im Hinblick auf unterschiedliche Aufgaben evaluiert. Als visuelle Hinweise setzten sie Höhe, Texturierung der Objekte sowie des räumlichen Bezugssystems (Wände), Schattenwurf (auf den virtuellen Boden), Perspektive und Bewegung ein. Die Testpersonen hatten zur Aufgabe, einen Körper möglichst exakt so zu positionieren, wie einen anderen Körper, der sich bereits in der Szene befand; dazu mußte der zu positionierende Körper rotiert und skaliert werden. Die Experimente ergaben, daß die visuellen Hinweise aufgabenbezogen wirken; sie können die Leistung der Testpersonen sowohl positiv als auch negativ beeinflussen, nur der Schattenwurf wirkte durchweg positiv. Die perspektivische Projektion machte sich bei der Orientierungs- und Skalierungsaufgabe negativ bemerkbar. Bewegung wurde von vielen Benutzern als störend empfunden, obwohl sie bei der Orientierung und Positionierung half. Das ist nach Meinung der Autoren darauf zurückzuführen, daß sich die Szene in dem Experiment kontinuierlich bewegte und die Bewegung nicht unter der Kontrolle der Benutzer war. Die Texturierung des Bezugsraums hatte keinen Einfluß; die der Objekte nur minimalen bei der Skalierungsaufgabe, was auf die Texturelemente zurückgeführt werden kann. Aus den Experimenten lassen sich keine generellen Empfehlungen für den Einsatz bestimmter Visualisierungshilfen ableiten, nur daß visuelle Hinweise an die augenblickliche Aufgabe dynamisch angepaßt werden sollten, um diese bestmöglich zu unterstützen. Wanger et. al. untersuchten nicht den Einfluß der 3-dimensionalen Ausgabe. Ein weiterer offener Punkt sind Untersuchungen typischer Interaktionen und Arbeitsabläufe in bestimmten Anwendungsfällen. Auch das sich ändernde Interaktionsverhalten mit zunehmender Erfahrung im Umgang mit einem System könnte die Art der Visualisierung beeinflussen. Problematisch bei solchen Untersu-

chungen ist der Umfang möglicher Kombinationen der Einflußfaktoren und das Ausschließen von Lernprozessen während der Untersuchungen, dem nur durch Gruppen von Testkandidaten begegnet werden kann, die die Experimente in jeweils umgekehrter Reihenfolge durchführen. Dann wiederum sind gruppenspezifische Unterschiede und Erfahrungswerte kaum auszuschließen.

Der von Wanger et al. uneingeschränkt empfohlene Schattenwurf wurde ursprünglich zur Steigerung der Realitätsnähe eingesetzt; räumliche Bezüge zwischen Objekten standen dabei nicht im Mittelpunkt des Interesses. Verschiedene Algorithmen wurden entwickelt, um Schatten zu berechnen [22], [35]. Die Berechnung von Schatten in komplexen Szenen stellt immer noch ein Performanzproblem dar, gerade wenn Objekte bewegt werden und Schatten auf andere Objekte werfen sollen. Erst im zweiten Schritt wurden Schatten benutzt, um räumliche Bezüge zu verdeutlichen und um mit dem schattenwerfenden Objekt zu interagieren. Herndon et al. [73] präsentieren, wie sich Schattenprojektionen auf statische, virtuelle Wände im Bereich 'scientific visualization' einsetzen lassen, um Translationen in einer Ebene vorzunehmen und verdeckte Objektteile durch Reflexion des Pickstrahls am Schatten selektieren zu können. In [12] wird ein Ansatz vorgestellt, der mittels Rand- und Kernschatten nicht nur die Position eines Objektes über der Ebene, sondern auch die Entfernung von der Ebene visualisiert. Dabei wird der Effekt ausgenutzt, daß Objekte, die weiter von der Ebene entfernt sind, einen kleineren Kernschatten werfen als Objekte, die der Ebene näher sind - vorausgesetzt die Objekte sind gleich groß. Die Berechnung bzw. Darstellung solcher Schatten ist aufwendig, so daß die Autoren nur mit wenig komplexen Szenen experimentiert haben.

2.5 Systemarchitekturen

Wie in Kapitel 1 dargelegt, subsumiert ein benutzer-zentriertes Modellierungssystem Aspekte von CAD, VR und CSCW. In der Vergangenheit wurde eine Vielzahl von Architekturvorschlägen für VR- und CAD-Systeme[3] im allgemeinen und Benutzungsoberflächen [50] im speziellen entwickelt. Im folgenden wird je ein Vertreter beider Gattungen vorgestellt, um die charakteristischen Unterschiede herauszuarbeiten und die Notwendigkeit eines integrativen Ansatzes für ein Architekturkonzept eines benutzer-zentrierten Modellierungssystems zu motivieren. Weiterhin erfolgt ein Überblick über Ansätze zur Kooperationsunterstützung in der Modellierung.

2.5.1 Das CAD-Referenzmodell

Das Projekt 'CAD-Referenzmodell' [2] schlägt eine Architektur für moderne, flexible CAD-Systeme vor. Die Architektur (siehe Abbildung 13a) gliedert sich in den Systemteil, den Anwendungsteil, die Komponente zur Handhabung anwendungsspezifischen Wissens sowie die Produktmodell-Management-Komponente. Der Systemteil stellt möglichen CAD-Anwendungen unterschiedliche Dienste zur Verfügung, so daß diese von immer wiederkehrenden Funktionalitäten entlastet werden. Zu diesen Diensten gehört das Benutzungsoberflächen- und auch das Kommunikationssystem. Der Anwendungsteil repräsentiert konkrete Anwendungen, die in den Systemrahmen integriert werden können. Bei den Anwendungen kann es sich um CAx-Anwendungen unterschiedlichen Typs handeln.

Im Zusammenhang mit der vorliegenden Arbeit spielt insbesondere die Gestaltung der Architektur der Benutzungsoberfläche - hier innerhalb des Systemteils angesiedelt - eine wichtige Rolle. Von

3. Jasnoch [82] stellt verschiedene Architekturkonzepte für CAD-Systeme gegenüber.

besonderer Bedeutung ist der Interaktions-Manager mit Eingabe-Komponente, Direkt-Manipulations-Komponente und Ereignisverarbeitungs-Komponente (siehe Abbildung 13b).

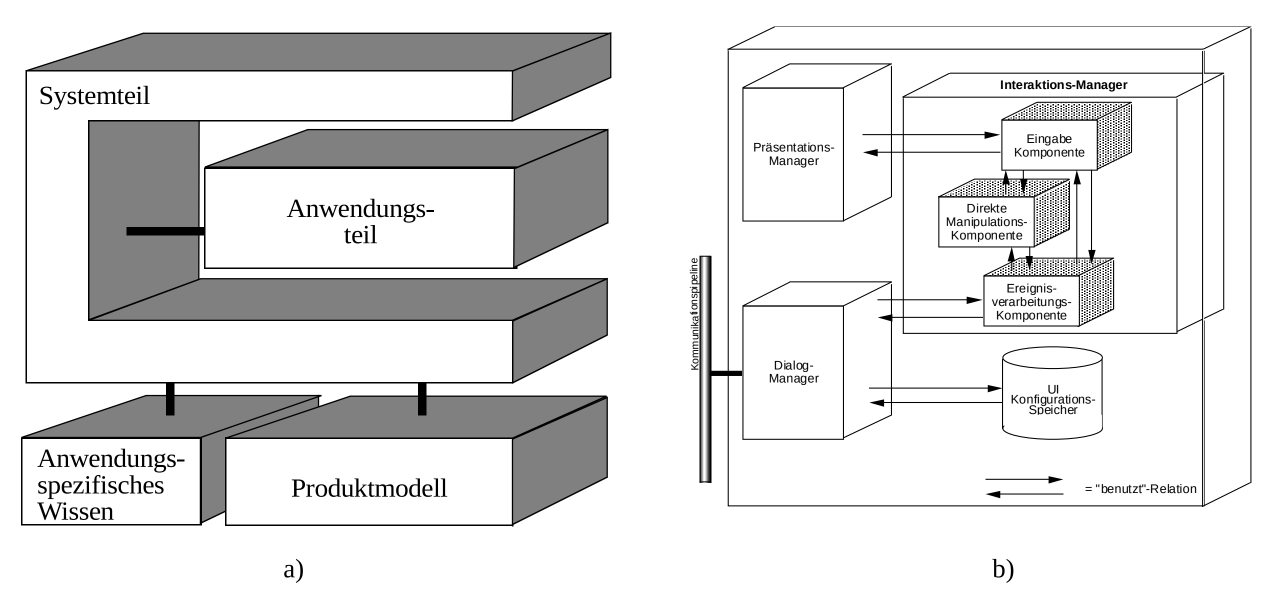


Abbildung 13: Architektur des CAD-Referenzmodells und des Interaktionsmanagers [2]

Es fällt auf, daß die Direkte-Manipulationskomponente keine direkte Verbindung zum Präsentations-Manager hat, der die graphische Ausgabe vornimmt. Weiterhin werden nur recht unspezifische Aussagen über die Funktionalität der Direkten-Manipulations-Komponente gemacht, und es fehlen Vorschläge und Verfahren, wie bestimmte Funktionen direkt-manipulativ ausgeführt werden können, obwohl bei diesem Architekturkonzept der Anwendungskontext, nämlich CAD, gegeben ist. Das CAD-Referenzmodell hat sich insbesondere den Aspekten der verteilten Produktentwicklung gewidmet, wobei ein Bestandteil das kooperative Modellieren war; Belange der Virtuellen Realität spiegeln sich kaum wider.

2.5.2 Systemarchitektur von VR-Systemen

Eine typische VR-Systemarchitektur ist in Abbildung 14 dargestellt. Zu erkennen ist die Szenendatenbank für den Import von CAD-Daten. Die importierten CAD-Daten werden zum Zwecke der performanten Darstellung trianguliert und 'VR-gerecht' aufbereitet; das eigentliche CAD-Modell steht zur Laufzeit nicht mehr zur Verfügung. Die VR-Systemarchitektur zeichnet sich durch Module zur Navigation und Interaktion sowie zur Gesten- und Kollisionserkennung aus. Zu erkennen ist auch die Unterstützung einer Vielzahl von Ein- und Ausgabegeräten. Ein Modul, das Modellierfunktionalität zur Verfügung stellt, ist in diesem Architekturkonzept nicht vorgesehen

Führt man sich das Einsatzspektrum heutiger VR-Systeme vor Augen, so ist dies nicht überraschend, werden VR-Systeme doch vernehmlich im Design Review und zur Ein- und Ausbausimulation eingesetzt. Ziel dabei ist es, die Anzahl zu bauender physikalischer Prototypen zu reduzieren und durch Untersuchungen am virtuellen Modell zu ersetzen (Digital Mock-Up). Design Review in VR, mit dem Ziel, ein Modell in seiner Komplexität möglichst realitätsnah darzustellen, um es inspizieren zu können und daraus Erkenntnisse über Mängel an der Konstruktion zu gewinnen, zeigt, wie wichtig und gewinnbringend alleine die Visualisierung sein kann. Ein anderes Anwendungsgebiet von VR-Systemen ist die Visualisierung von und Interaktion mit Simulationsdaten. Dabei kann es sich sowohl um Daten einer kinematischen Simulation als auch um Daten z.B. einer Strömungssimulation handeln. Die Interaktivität von VR-Systemen erweitert dabei die Möglichkeiten existierender *scientific visualization*-Systeme. Die genannten

verschiedenartigen Daten werden im VR-System durch ein zentrales Objektverwaltungsmodul gehandhabt.

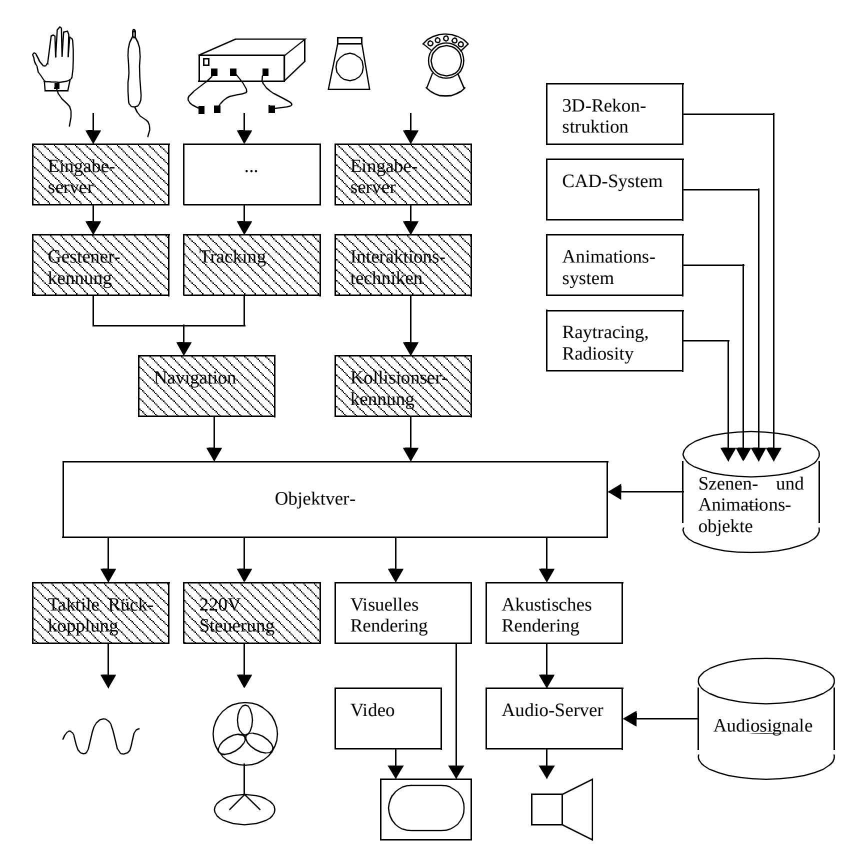


Abbildung 14: Architektur eines VR-Systems nach Felger [57]

2.5.3 Ansätze zum kooperativen Modellieren

Systeme und Ansätze zur Kooperationsunterstützung - auch Groupware genannt - haben in den vergangenen Jahren dazu geführt, daß Benutzer von CAD-Systemen miteinander kooperieren können. Die wichtigsten Ansätze hierzu, nämlich das Application Sharing sowie die Kopplung von Systemen mittels Nachrichtenaustausch, sollen im folgenden mit ihren Vor- und Nachteilen kurz vorgestellt werden.

Application-Sharing-Tools erlauben das kooperative Arbeiten mit CAD-Systemen, allerdings dürfen die Benutzer nicht zur gleichen Zeit agieren, sondern müssen im Besitz des Aktionsrechts sein, um eine Aktion ausführen zu können. Zu jedem Zeitpunkt kann immer nur ein Benutzer das Aktionsrecht besitzen. Application-Sharing-Systeme sorgen dafür, daß die Bildschirmausgabe eines Anwendungssystems an verschiedene, örtlich verteilte Benutzer ausgegeben wird. Eingaben können von den Benutzern abwechselnd durchgeführt werden, wobei das Application-Sharing-Tool die Ereignisse an die ’gesharte’ Instanz des Anwendungssystems weiterleitet und diese dort ausgewertet werden (siehe Abbildung 15 links). Durch die reine Vervielfältigung der Bildschirmausgabe kann nicht applikationsübergreifend verteilt gearbeitet werden, d.h. kooperatives Arbeiten zwischen einer CAD- und einer VR-Applikation ist nicht möglich; auch müssen sich die Benutzer die Sicht auf das Modell teilen.

Beim nachrichten-basierten Ansatz wird kooperatives Modellieren durch den Austausch von Nachrichten mit entfernten Systeminstanzen realisiert. Eine Nachricht, z.B. ein Modellierungskommando, wird über Netzwerk an mehrere Instanzen eines CAD-Systems übermittelt (siehe

Abbildung 15 rechts). Der nachrichten-basierte Ansatz bietet gegenüber dem Application-Sharing Vorteile im Hinblick auf Flexibilität, Performanz und Netzwerklast, erfordert aber explizit eine Konsistenzsicherung und Aktionsrechtverwaltung seitens des nachrichtenaustauschenden Systems. Nachrichten in Form von Modellierungskommandos sind nicht in der Lage, die interaktiven Prozesse der Partner abzubilden. Im Gegensatz zum Application Sharing ist es mit einem nachrichten-basierten Ansatz möglich, unterschiedliche Systeme miteinander kommunizieren zu lassen.

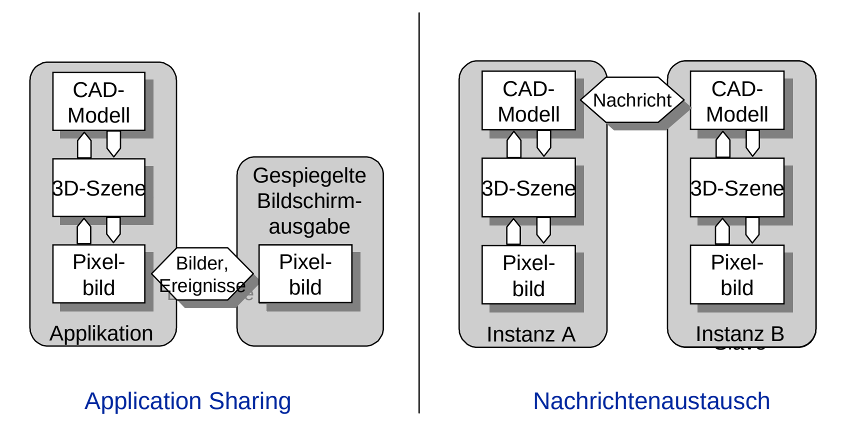


Abbildung 15: Gegenüberstellung von Application Sharing und Nachrichtenaustausch

Dietrich et. al. [49] stellen mit ’TOBACO’ einen CORBA-basierten Ansatz vor, bei dem alle Komponenten eines Modellierungssystems als verteilte Objekte realisiert sind. Die einzelnen Module können prinzipiell auf unterschiedlichen Rechnern laufen. Ein *object request broker* (ORB) sorgt für den Aufruf des entfernten Objektes. Dieses Konzept der kompletten Verteilung eines Systems wird mit einem gewissen Overhead bezahlt, so daß Abstriche bezüglich der erzielbaren Performanz, Interaktivität und Echtzeitfähigkeit gemacht werden müssen. Die interaktiven Prozesse der Partner werden nicht übertragen; Direktmanipulation steht nicht im Mittelpunkt der Betrachtungen.

Carlsson und Hagsand [33] stellten mit DIVE erstmals ein verteiltes Multiuser-VR-System vor, in dem mehrere Benutzer ortsunabhängig interagieren konnten. DIVE basiert ebenfalls auf dem Austausch von Nachrichten zwischen Instanzen eines VR-Systems. Eine immersive, verteilte Modellierungsanwendung auf Basis eines Nachrichtenaustauschs wird in [39] beschrieben.

2.6 Zusammenfassung und Handlungsbedarf

In diesem Kapitel wurde der aktuelle Stand der Technik der von der Aufgabenstellung dieser Arbeit berührten Themengebiete dargestellt. Auch wurden die Grundlagen gelegt und Begriffe geklärt, die im Verlauf der vorliegenden Arbeit von Bedeutung sind. Darüber hinaus wurden 3D-Ein- und Ausgabegeräte vorgestellt und bezüglich ihrer Eignung im Modellierungsprozeß bewertet. Auf den Stand der Technik wird im folgenden nochmals zusammenfassend eingegangen, die Mankos im Hinblick auf die Aufgabenstellung dargestellt und daraus der Handlungsbedarf für diese Arbeit abgeleitet.

Aus dem Abschnitt über Systemarchitekturen und Ansätze zur Kooperationsunterstützung geht hervor, daß kooperatives Modellieren an einem verteilten CAD-System bislang entweder mittels Application-Sharing oder Nachrichtenaustausch unterstützt wird. In verteilten, immersiven Umgebungen kommt ebenfalls Nachrichtenaustausch zum Einsatz. Kooperatives Arbeiten zwischen einem CAD-System und einer immersiven Umgebung wird bislang nicht unterstützt.

Die existierenden Arbeiten lassen erkennen, daß noch keine übergreifende Betrachtung, die CAD, VR und Kooperation gleichermaßen einbezieht, stattgefunden hat. Es fehlen Vorschläge, wie eine Direkte-Manipulations-Komponente aussehen und welche Verfahren zur Interaktion und Visualisierung sie anbieten muß, um den Anforderungen von CAD und VR gerecht zu werden. Dasselbe gilt für die Kommunikationskomponente, die in einem benutzer-zentrierten Modellierungssystem mit neuen Anforderungen konfrontiert ist. Es gilt daher ein Architekturkonzept für Benutzer-zentrierte Modellierungssysteme zu konzipieren, welches in der Lage ist, die Anforderungen von CAD und VR zu erfüllen und geeignete Kommunikationsmechanismen zur Kooperationsunterstützung bietet, um unterschiedliche Instanzen eines solchen Systems zu koppeln. Die Konzeption einer solchen Systemarchitektur erfolgt im nächsten Kapitel.

Der Stand der Technik zur Interaktion mit 3D-Eingabegeräten zeigt, daß diese ein erhöhtes Maß an Intuitivität und natürlichem Umgang bieten. Aus den existierenden Arbeiten geht hervor, daß 3D-Eingabegeräte im Konstruktionsprozeß Vorteile versprechen, die bislang allerdings nur für den unpräzisen Entwurf in der Konzeptionierungsphase umgesetzt werden konnten. Hingegen mangelt es an geeigneten Interaktionstechniken für 3D-Eingabegeräte, die schnelles mit präzisem Arbeiten im graphisch-interaktiven 3D-Dialog verbinden, und somit den speziellen Anforderungen in der Konstruktion Rechnung tragen. Basierend auf den Anforderungen der Modellierung und den Möglichkeiten von 3D-Eingabegeräten werden in Kapitel 4 neuartige Interaktionstechniken sowie unterstützende Methoden und Verfahren entwickelt, die erstmals schnelles und präzises Arbeiten mit 3D-Eingabegeräten ermöglichen. Dabei findet nicht nur die Bauteilmodellierung, sondern auch die Zusammenbausimulation Berücksichtigung.

Der Abschnitt über die Visualisierung stellt dar, daß der Mensch Form- und Tiefenhinweise aus einer Vielzahl von Informationen bezieht. Viele dieser Merkmale lassen sich heute schon in computer-generierten Bildern nachbilden. In 3D-CAD-Systemen kommen diese Präsentationsmittel allerdings oft nicht zum Einsatz. Die existierenden Untersuchungen zur Rolle der Visualisierung zeigen, welchen Einfluß diese auf die Interaktionsgeschwindigkeit haben kann. Dabei wurde der Einfluß stereoskopischer Darstellung und der Anwendungskontext CAD bislang noch nicht hinreichend beachtet. Daß das Arbeiten mit 3D-Eingabegeräten geeignete Visualisierungstechniken erfordert, ist intuitiv klar. Es stellt sich jedoch die Frage, wie das Modell und der Konstruktionsraum dargestellt werden sollte, um eine effiziente 3D-Interaktion zu begünstigen und welche Wechselwirkungen zwischen Interaktion und Visualisierung bestehen. Zu diesem Zweck werden in Kapitel 4 Visualisierungstechniken und visuelle Hinweise vorgeschlagen, die die Form- und Tiefenwahrnehmung verbessern und die Interaktion in 3D durch Visualisierung der laufenden Benutzeraktion sowie der Interaktionsmöglichkeiten erleichtern.

Viele Ansätze aus der Literatur werden von den Autoren als sehr intuitiv bezeichnet und nehmen große Zeitvorteile im Vergleich mit anderen Ansätzen für sich in Anspruch, ohne daß eine Evaluierung erfolgt oder die Rahmenbedingungen des Vergleichs detailliert beschrieben werden. Daher widmet sich Kapitel 5 der Evaluierung der im Rahmen dieser Arbeit entwickelten Verfahren und stellt deren theoretische Vorteile und praktischen Nutzen dar.

3 Architekturkonzept für benutzer-zentrierte 3D-Modellierungssysteme

In diesem Kapitel wird eine Systemarchitektur für benutzer-zentrierte 3D-Modellierungssysteme konzipiert. Dazu werden zunächst die Anforderungen, die ein solches System erfüllen soll, dargelegt, um darauf aufbauend das Systemkonzept zu entwickeln. Den Schwerpunkt des Systemkonzeptes bilden Aspekte der Benutzungsoberfläche - auch im Hinblick auf kooperatives Arbeiten. Die aus der Sicht der Benutzungsoberfläche wichtigen Komponenten werden detailliert und Implementierungsaspekte werden diskutiert.

3.1 Anforderungen an ein benutzer-zentriertes 3D-Modellierungssystem

Ein benutzer-zentriertes 3D-Modellierungssystem muß die allgemeinen Anforderungen an benutzer-zentrierte Systeme sowie die besonderen Anforderungen der Modellierung erfüllen. Daher seien zunächst nochmals die allgemeinen Anforderungen benutzer-zentrierter Systeme aufgeführt:

- Die wahrnehmungspsychologischen sowie motorischen Fähigkeiten und Einschränkungen des Menschen sollen beachtet werden, d.h. geeignete Visualisierung und direkt-manipulative Interaktion unter Beachtung der Reiz-Reaktions-Korrespondenz sind vorzusehen.
- Die Arbeitsaufgaben des Benutzers sollen auf effiziente Art und Weise unterstützt werden.
- Die Kooperation mit anderen Benutzern soll möglich sein.
- Eine flexible Nutzung, die den Benutzer nicht unnötig einschränkt, soll möglich sein, d.h. das von dem System gebotene Maß an Immersion und Kooperation soll sich nach der zu bearbeitenden Aufgabe richten.
- Sie sollen sich kontext-sensitiv verhalten und an die sich ändernden Benutzeranforderungen anpassen.
- Sie sollen selbsterklärend sein und vom Benutzer leicht inspiziert und verstanden werden können.
- Ihre Benutzung soll die Kreativität anregen anstatt zu frustrieren.

Aus der Sicht der Modellierung ergeben sich hauptsächlich Anforderungen hinsichtlich der Funktionalität eines solchen Systems:

- Die Erzeugung und Manipulation von Modellen muß möglich sein.
- Modellierungsoperationen, wie z.B. Boolsche Verschneidungen und Sweeping, müssen zur Verfügung gestellt werden.
- Die besonderen Anforderungen bezüglich Ergonomie und Präzision müssen beachtet werden.
- Die Semantik der Konstruktionsobjekte sollte berücksichtigt werden.

Bei dem Versuch, die genannten Anforderungen auf ein reales System abzubilden wird klar, daß Modellierungsfunktionalität um Aspekte der Virtuellen Realität hinsichtlich Interaktion und Visualisierung und Ansätze zur Kooperationsunterstützung erweitert werden muß. In Anbetracht dessen, ergibt sich als sinnvoller Ansatz für die Konzeption einer Systemarchitektur eines benutzer-zentrierten 3D-Modellierungssystems die Erweiterung einer CAD-Systemarchitektur [151]. Von besonderer Bedeutung sind dabei die Implikationen, die die Forderung nach Direktmanipulation nach sich zieht.

3.2 Konzeption der Systemarchitektur

Die Grundlage der Konzeption einer Systemarchitektur für benutzer-zentrierte 3D-Modellierungssysteme bilden bekannte Architekturkonzepte aus dem CAD-Bereich [2], [82]. Diese Architekturkonzepte gilt es so zu erweitern, daß die o.g. Anforderungen umgesetzt werden können.

Die Erweiterung basiert auf folgenden grundlegenden Ideen:

- Die verschiedenen Systemkomponenten halten dedizierte Objektrepräsentationen, die den Anforderungen an die jeweilige Komponente entsprechen. Beispielsweise enthält die graphische Ein-/Ausgabekomponente zum Zwecke der schnellen Visualisierung nur ein trianguliertes Modell, wohingegen das mathematisch präzise CAD-Modell nur in der Modellierungskomponente gehalten wird.
- Ein zentraler Objektverwalter sichert die Konsistenz zwischen den korrespondierenden Objektrepräsentationen.
- Der Benutzer ist in die Interaktionsschleife eingebettet (*user on the loop*-Konzept).
- Die graphische Ein-/Ausgabekomponente ist mit Modellierungsfunktionalität angereichert, um direkt-manipulative Modellierung zu ermöglichen und den Zugriff auf das präzise CAD-Modell zu minimieren.
- Eine integrierte CSCW-Komponente bietet per online-Kopplung die Möglichkeit zum kooperativen Arbeiten zwischen Systeminstanzen, die bezüglich der aktiven Ein-/Ausgabegeräte unterschiedlich konfiguriert sein können; simultanes und direkt-manipulatives Arbeiten soll unterstützt werden.
- Das gewünschte Maß an Immersion wird durch die Anbindung verschiedener Ein-/Ausgabegeräte erreicht.

Basierend auf diesen Überlegungen wurde im Rahmen dieser Arbeit die in Abbildung 16 dargestellte Systemarchitektur konzipiert, deren Module sowie deren Aufgaben im folgenden kurz erläutert werden. Den Modulen GraphikManager, HistoryManager und NetzwerkManager ist ein eigenes Unterkapitel gewidmet, in dem Implementierungsaspekte diskutiert und die in diesen Modulen realisierten neuartigen Ansätze beschrieben werden. Im Hinblick auf ein benutzer-zentriertes System bilden Komponenten der Benutzungsoberfläche den Schwerpunkt der Systemarchitektur, so daß diese in einer feineren Granularität dargestellt sind.

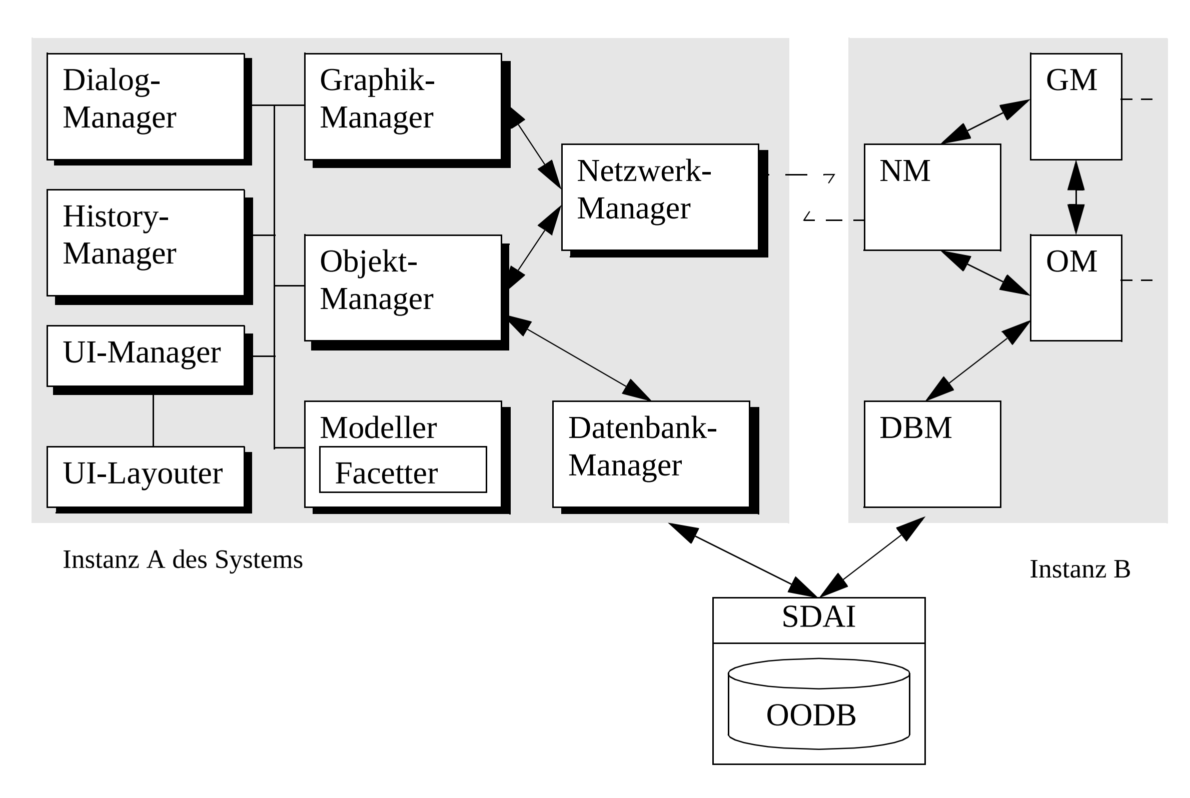


Abbildung 16: Architekturkonzept für ein benutzer-zentriertes Modellierungssystem

Die Aufgabe des **ObjektManagers** (OM) als zentrale Verwaltungsinstanz ist es, die dedizierten Repräsentationen jedes Objektes, die im System zur Befriedigung der Anforderungen an Präzision und Performanz gehalten werden, zu verwalten und die Konsistenz zwischen diesen korrespondierenden Objektrepräsentationen zu sichern. Der ObjektManager übernimmt die Verteilung von Aktionen innerhalb des Systems, unabhängig davon, ob sie von dem GraphikManager kommen oder über das Netzwerk eintreffen. So können beispielsweise Boolsche Operationen direkt manipulativ vom GraphikManager vorgenommen werden, um erst nach Abschluß der Interaktion die präzise Repräsentation des CAD-Modells im Modeller zu aktualisieren. Der ObjektManager ist für die Koordination und Propagierung derartiger Änderungen im System verantwortlich und gewährleistet damit die Konsistenz der Objektrepräsentationen.

Der **GraphikManager** (GM) handhabt die graphisch-interaktive Ein- und Ausgabe und unterstützt eine breite Palette an Ein- und Ausgabegeräten. Er realisiert geeignete 3D-Interaktionstechniken und enthält die graphische Repräsentation aller Objekte. Diese stellt ein Objekt als tesselliertes Modell dar und ermöglicht somit eine schnelle Visualisierung. Das tessellierte Modell ist auch die Basis für beschleunigte Modellierungsoperationen während direkter Manipulationen.

Der **HistoryManager** verwaltet die Konstruktionshistorie und bietet die Möglichkeit, Varianten und Versionen zu erstellen. Zu diesem Zweck wurde eine erweiterte CSG-Datenstruktur entwickelt, der HistoryGraph, der in dem Abschnitt zum HistoryManager (siehe Kapitel 3.4) beschrieben wird.

Die Aufgabe des **NetzwerkManagers** (NM) ist es, kooperatives Arbeiten zwischen unterschiedlichen Systeminstanzen zu ermöglichen. Bei der Konzeption dieser Komponente sind den prinzipiell widersprüchlichen Anforderungen nach geringer Netzwerklast und Echtzeitfähigkeit im kooperativen Betrieb Rechnung zu tragen. Diesen Anforderungen wurde durch Kopplung der Instanzen mittels Nachrichtenaustausch auf verschiedenen Ebenen begegnet (siehe Kapitel 3.5).

Daher besteht in Abbildung 16 sowohl eine Verbindung zum Graphik- als auch zum ObjektManager.

Der **Modeller** enthält die mathematisch genaue Repräsentation aller Objekte - das eigentliche CAD-Modell. Seine Aufgabe besteht darin, die benötigte Modellierungsfunktionalität auf der Ebene der präzisen Modellrepräsentation bereit zu stellen. Der **Facetter** tesselliert das präzise CAD-Modell zur schnellen Visualisierung durch den GraphikManager. Da das Modell permanent zur Verfügung steht, kann für Visualisierungen oder Berechnungen die Tessellierungsgenauigkeit ohne weiteres zur Laufzeit variiert werden; dies stellt für heutige VR-Systeme häufig noch ein Problem dar, da sie als Eingangsdaten bereits tessellierte Modelle erhalten.

Der **UI-Manager** realisiert die für die Bedienung am herkömmlichen Arbeitsplatz benötigte 2D-Benutzungsoberfläche. Für eine kontext-sensitive arbeitsaufgabengerechte Konfiguration der Benutzungsoberfläche stellt der **UI-Layouter** die benötigte Funktionalität zur Verfügung [3], [52].

Der **DialogManager** verwaltet den Dialogstatus des Systems. Besonders hervorzuheben sind die Anforderungen, die die multi-modale Eingabe und der Mehrbenutzerbetrieb stellen. Bei multimodaler Eingabe [24] müssen Ereignisse, die von unterschiedlichen Eingabegeräten herrühren, z.B. Spracheingabe und 3D-Eingabe, synchronisiert und entsprechend ausgewertet werden. Im kooperativen Betrieb müssen die Aktionen, die von Benutzern verteilter Systeminstanzen stammen, gehandhabt werden.

Zur Verwaltung aller Produkt-relevanten Daten wird ein STEP-basierter [79] Ansatz auf Basis einer objekt-orientierten Datenbank vorgeschlagen. Der **Datenbank-Manager** (DBM) hat die Aufgabe, die über SDAI (standard data access interface) [80] zugreifbaren STEP-konformen Daten auf die internen Datenstrukturen abzubilden [25]. Um der Anforderung nach kooperativem Einsatz gerecht zu werden, muß auf die Datenbank verteilt zugegriffen werden können. Das STEP-konforme SDAI erfüllt zur Zeit nicht die Anforderungen nach konkurrierendem Zugriff. Mit einer Erweiterung von SDAI [81] kann diese Einschränkung überwunden werden.

Das vorgestellte Architekturkonzept verbindet Eigenschaften von CAD- und VR-Systemen und sieht Kommunikationsunterstützung als inhärenten Bestandteil an. Damit sind die Basisanforderungen, die benutzer-zentrierte Systeme stellen, abgedeckt. Zugleich können viele der in der Einleitung genannten Probleme beim heutigen Einsatz von VR- und CSCW-Systemen in der Produktentwicklung überwunden werden. Insbesondere im Bereich des Datenaustauschs ergeben sich folgende Möglichkeiten und Vorteile:

- Das CAD-Modell steht jederzeit direkt zur Verfügung, so daß Berechnungen - wenn nötig - auf dem mathematisch präzisen Modell durchgeführt werden können.
- Kooperation kann zwischen einer (semi-)immersiven Instanz des Systems und einer Instanz, die am herkömmlichen Arbeitsplatz bedient wird, stattfinden.
- Interaktionen, wo immer sie stattgefunden haben, ob innerhalb einer VR-Instanz oder am Arbeitsplatz oder am Virtuellen Tisch, können direkt zu dem CAD-Modell im Modeller propagiert werden, was einem bidirektionalen Datenaustausch gleichkommt.
- Interaktionen können im Sinne des CAD-Modells semantisch korrekt durchgeführt werden.

Im folgenden werden die Komponenten der Systemarchitektur detaillierter beschrieben, die zur Schaffung dieser Möglichkeiten maßgeblich beitragen.

3.3 Der GraphikManager

Der GraphikManager (GM) bildet das Kernmodul der Integration von VR-Technologie in das Systemkonzept. Hinsichtlich 3D-Eingabeverarbeitung, 3D-Ausgabesteuerung und 3D-Interaktionstechniken geht er deutlich über das von CAD-Systemen gewohnte Maß hinaus. Neben der graphischen Repräsentation hält der GM Interaktionstechniken und Manipulationsmöglichkeiten für verschiedene Objekttypen bereit, verwaltet topologische sowie semantische Informationen zu den Objekten und die Zugehörigkeit von Facetten zu Flächen und Objekten im CAD-Modell. Außerdem bietet er Unterstützung für CSCW-typische Funktionen zur Erhöhung der Telepräsenz, wie einen *remote cursor* und eine Repräsentation der entfernten Benutzer (Avatare).

Der GraphikManager gliedert sich in die Module: Interaktionskomponente, Präsentationskomponente und Direkte-Manipulationskomponente. Zur Verbesserung des Antwortzeitverhaltens wird durch die Direkte-Manipulationskomponente eine enge Kopplung von Interaktion und Visualisierung realisiert (siehe Abbildung 17). Der Benutzer ist direkt in die Interaktions- und Visualisierungsschleife eingebettet. Das Prinzip des 'user-in-the-loop' ermöglicht ein hohes Maß an Interaktivität. In herkömmlichen CAD-Systemen hingegen wird oft folgendermaßen verfahren: Entgegennehmen eine Benutzerinteraktion, Auswertung der Aktion im Modellierkern und schließlich Präsentation des Ergebnisses. Der hier vorgestellte Ansatz umgeht durch Realisierung von Modellierungsfunktionalität auf den tessellierten Objekten im GraphikManager die zeitaufwendige Aktualisierung des präzisen CAD-Modells im Modellierkern während direkter Manipulationen, was eine höhere Interaktivität ermöglicht.

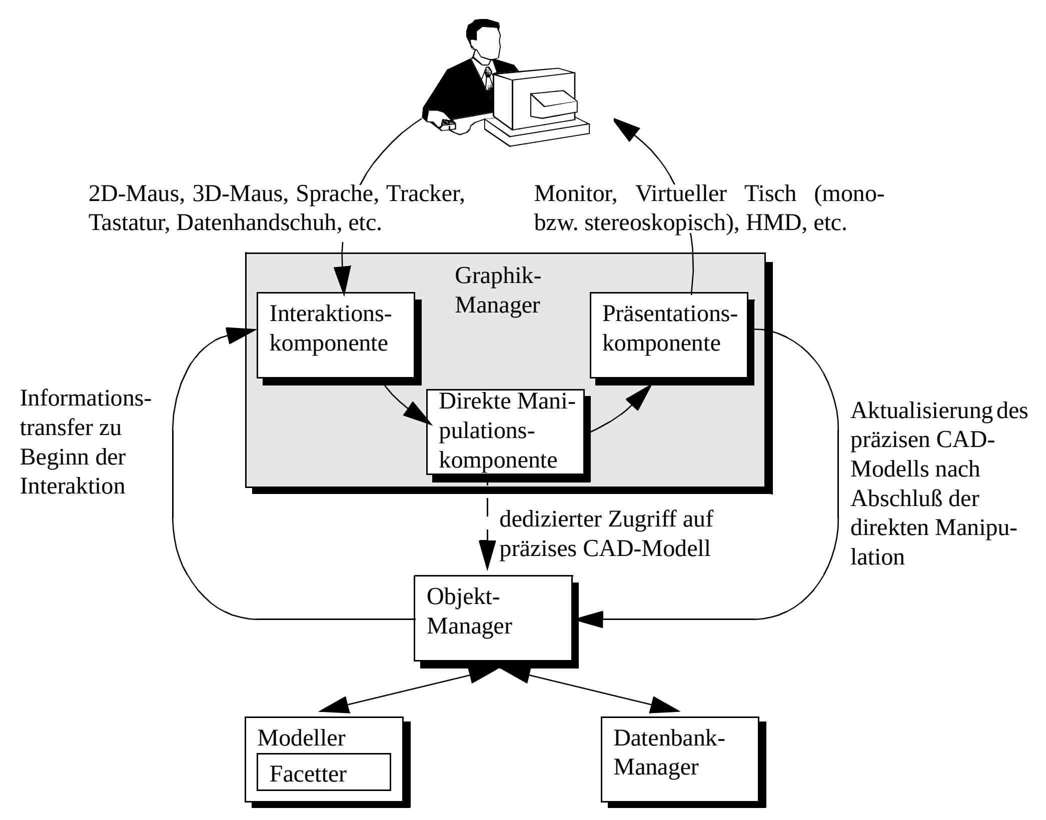


Abbildung 17: Direkte Manipulation durch Modellierungsfunktionalität im GraphikManager

Abbildung 17 veranschaulicht den Ablauf einer direkt-manipulativen Interaktion. Zu Beginn einer direkten Manipulation erfolgt ein dedizierter Zugriff auf das präzise CAD-Modell, um die genaue Pick-Information und das gepickte topologische Element anzufordern. Dann wird die Aktion direkt-manipulativ durchgeführt (siehe unten: Direkte-Manipulationskomponente). Anschließend

werden die Parameter aus der Interaktion an den ObjektManager weitergegeben, der seinerseits eine konsistente Aktualisierung des CAD-Modells im Modeller initiiert. Abschließend wird das aktualisierte CAD-Modell unter den benutzer-definierten Facettierungsungenauigkeiten (partiell) neu facettiert, so daß die Repräsentation des Modells im GM wieder der Repräsentation im Modeller entspricht.

Die **Interaktionskomponente** nimmt Ereignisse von der Tastatur, 2D- und 3D-Eingabegeräten (3D-Maus, Data Glove), Trackingeinheiten sowie Sprachkommandos entgegen und verarbeitet diese.

Die **Präsentationskomponente** übernimmt, basierend auf dem Graphik- und Fenstersystem, die mono- bzw. stereoskopische Ausgabe auf einem herkömmlichen Bildschirm, Virtuellen Tisch oder einer Großbildprojektion.

Je nach Ausprägung der jeweiligen Systeminstanz müssen die Interaktions- und die Präsentationskomponente auf die unterschiedlich konfigurierten Ein- und Ausgabegeräte reagieren. Im immersiven Modus sind z.B. Funktionen, die sich im konventionellen Modus hinter 2D-Menüeinträgen verbergen, über Sprachkommandos ansprechbar.

Die **Direkte-Manipulationskomponente** (DMK) ist mit Modellierfunktionalität angereichert[1], um Echtzeitfähigkeit während häufig benutzter Modellieroperationen zu erreichen. Die DMK erlaubt:

- Objekte schnell und semantisch korrekt zu erzeugen und zu modifizieren.
- Boolsche Operationen rasch zu berechnen.
- Direkt-manipulatives Sweeping durchzuführen.

Mit Hilfe von geeigneten Verfahren können CAD-typische Objektmanipulationen im GM gehandhabt werden. Das CAD-Modell im Modeller wird erst nach Abschluß der direkten Manipulation und nicht währenddessen aktualisiert, was einen grundlegenden Geschwindigkeitsvorteil bietet.

Anzumerken bleibt, daß CAD-Funktionen existieren, die soviel Parameter erfordern, daß ein direkt-manipulatives Vorgehen nicht mehr möglich oder sinnvoll ist, z.B. das Sweeping entlang eines regel-basierten Sweep-Pfades. Eine Unterstützung aller Modellierungsoperationen durch die Direkte-Manipulationskomponente käme der Verlagerung der Funktionalität des Modellierkerns in diese Komponente gleich, was im Hinblick auf Kriterien des Software-Engineerings, wie Modularität, sicher nicht erstrebenswert ist. Weiterhin ist zu beachten, daß die im GM angesiedelten Modellierungsfunktionalitäten ihren Geschwindigkeitsvorteil aus der Tatsache beziehen, daß sie auf einem tessellierten Modell arbeiten und nicht ständig mit der vollen Genauigkeit des zugrundeliegenden CAD-Modells. Dies ist für den Zeitraum der direkt-manipulativen Interaktion völlig ausreichend. Anschließend wird die Manipulation an das CAD-Modell propagiert, so daß im CAD-Kern wieder ein exaktes und aktualisiertes Modell vorliegt. Während bestimmter direkt-manipulativer Interaktionen kann es aufgrund der Approximation vorübergehend zu visuellen Artefakten kommen.

1. Tendenzen, Graphiksysteme mit CAD-Funktionalität anzureichern, finden sich auch in Tools, wie z.B. OpenGL Optimizer [109].

3.3.1 Effiziente Objekterzeugung und -modifikation

Zur schnellen Objekterzeugung und -manipulation realisiert die DMK Interaktionstechniken für 2D- und 3D-Primitive, d.h. Grundkörper können in Echtzeit - und mit dem entsprechenden visuellen Feedback - kreiert und modifiziert werden. Die Interaktionstechniken (siehe Kapitel 4) beachten die Semantik der Objekte. Diskretisierte Interaktionen und die Topologie-basierte eingeschränkte Modifikationstechnik (siehe Kapitel 4.2.6) ermöglichen effizientes und präzises Modellieren mit 3D-Eingabegeräten. Darüber hinaus können die Interaktionstechniken parametrisiert werden, um nur erlaubte Modifikationen zuzulassen. Die Parametrisierung kann z.B. aus einem feature-basierten Modell abgeleitet werden.

3.3.2 Schnelle Boolsche Operationen

Boolsche Operationen werden in herkömmlichen CAD-Systemen im Modellierkern durchgeführt, das Ergebnis wird tesselliert und anschließend zur Darstellung gebracht. Diese Prozedur erlaubt es nicht, Boolsche Operationen dynamisch in Echtzeit durchzuführen. Alternativ bieten sich spezielle, schnelle Verfahren auf dem tessellierten Modell an, um Boolsche Operationen durchzuführen bzw. zu simulieren. Drei Verfahren, um dies zu tun, sind aus der Literatur bekannt:

- Binary Space Partitioning Trees [106]
- Boolsche Operationen im Bildraum unter Nutzung von *depth buffer* und *stencil buffer* [179]
- Convex Differences Aggregates (CDAs) [118]

Bei dem ersten Ansatz wird das tessellierte Modell in einen Binary Space Partitioning Tree (BSPT) überführt, auf dem eine Boolsche Operation schneller als auf der mathematisch genauen Präsentation des Modells ausgeführt werden kann. Binary Space Partitioning Trees sind eine Halbraumdarstellung, wobei die Halbräume durch Halbebenen definiert werden. BSPTs wurden ursprünglich zur Tiefensortierung und zum schnellen Rendering eingesetzt [65]. BSPTs eignen sich auch, um Schattenvolumen zu berechnen und somit einen Schattenwurf zwischen Objekten zu erzeugen [35].

Das Bildpuffer-basierte Verfahren nutzt den Tiefenpuffer (z-Buffer) und den *stencil buffer* moderner Graphik-Hardware, um die Berechnung des Ergebnisses einer Boolschen Operation durch geschicktes Rendering der dem Benutzer zu- bzw. abgewandten Flächen der involvierten Objekte zu simulieren. Im Unterschied zu BSPTs arbeitet es im Bildraum und nicht im Objektraum. Wiegand [179] vergleicht diesen Ansatz mit der BSPT-basierten Methode von Thibault und Naylor und kommt dabei zu erheblichen Zeitvorteilen zugunsten der Bildpuffer-basierten Methode.

Die dritte Methode arbeitet änlich wie die BSP-Trees auf Polyedern, die aus dem CAD-Modell durch Tessellierung resultieren. Ein CDA repräsentiert einen Polyeder als Vereinigung einer positiven konvexen Zelle mit einer Menge von negativen konvexen Zellen. Die von Rappoport [118] für Boolsche Operationen auf CDAs gemessenen Zeiten, lassen den Ansatz im Vergleich mit dem Bildpuffer-basierten Verfahren für die interaktive Modellierung als weniger geeignet erscheinen.

Im Rahmen dieser Arbeit wurden die beiden erstgenannten Verfahren implementiert. BSPTs zeigten bei komplexen CAD-Modellen numerische Instabilitäten, so daß der Ansatz nicht weiter verfolgt wurde [116]. Der Bildpuffer-basierte Ansatz zeigte ein robustes und hinreichend schnelles

Verhalten. Wie aus einer 3D-Interaktion das automatische Ausführen einer Boolschen Operation abgeleitet werden kann, wird im Rahmen der Interaktionstechniken in Kapitel 4.2.4 erläutert.

3.3.3 Direkt-manipulatives Sweeping

Die Darstellung von Geometrie in ihrer approximierten Form durch den GraphikManager ermöglicht die schnelle Berechnung von einfachen Sweepings. In Kombination mit einem 3D-Eingabegerät ist es möglich, planare Flächen entlang eines stückweise linearen 3D-Sweep-Pfades in Echtzeit zu extrudieren. Das entstehende Volumen wird im GraphikManager von der Direkten-Manipulationskomponente errechnet und von der Präsentationskomponente dargestellt. Da alle planaren Flächen durch Polygone approximiert werden, können auch kreisförmige und elliptische Flächen interaktiv extrudiert werden. Rigides Sweeping ist ebenso möglich wie nicht-rigides Sweeping. Da im allgemeinen Sweep-Operationen sehr kompliziert sein können und z.T. viele Parameter erfordern, die schwer in einem direkt-manipulativen Prozeß spezifiziert werden können, ist Echtzeitfähigkeit nur für die genannten Fällen sinnvoll zu realisieren. Trotz dieser Einschränkung eröffnen auch hier 3D-Eingabegeräte neue Möglichkeiten: so können Sweeppfade direkt 3-dimensional eingegeben werden und der Benutzer kann interaktiv sein Sweep-Körper generieren, wobei er ein Echtzeit-Feedback während der Erzeugung dargestellt bekommt. In Kombination mit echtzeit-fähigen Boolschen Operationen ergibt sich die Möglichkeit zum additiven bzw. subtraktiven Sweeping (vgl. Kapitel 4.2.5).

3.4 Der HistoryManager

Der GraphikManager stellt - wie gerade beschrieben - in Verbindung mit dem Modeller Funktionalität zur Geometrieerzeugung bereit. Im Bereich des mechanischen CAD ist es aufgrund der Häufigkeit von Anpaß- und Variantenkonstruktionen wichtig, auf die Konstruktionshistorie zurückgreifen und dort Veränderungen vornehmen zu können, um ein Bauteil zu modifizieren oder eine Variante bzw. Version eines existierenden Bauteils zu erzeugen.

Zu diesem Zweck sieht das Systemkonzept einen HistoryManager vor, dessen Aufgabe es ist, die Konstruktionshistorie zu verwalten. Da sich die Funktionalität eines benutzer-zentrierten Modelliersystems nicht auf die reine CSG-Modellierung (constructive solid geometry) beschränken soll, ist eine Datenstruktur notwendig, die mehr Flexibilität bietet als der klassische CSG-Baum.



Abbildung 18: Ein CSG-Baum und das resultierende Volumenmodell

Constructive solid geometry (CSG) [120] stellt einen Ansatz zur Volumenmodellierung dar. Dabei werden Grundprimitive mittels Boolscher Operationen zu komplexeren Modellen kombiniert. Die Grundprimitive können mittels Transformationen im Raum plaziert werden. Ein Modell wird in Form eines CSG-Baumes beschrieben (siehe Abbildung 18). Die Wurzel des Baumes beschreibt das resultierende Objekt. In den Knoten stehen nur Boolsche Operationen und Transformationen. Die Knoten enthalten i.d.R. keine Zwischenergebnisse, so daß das Modell stets neu evaluiert

werden muß, wenn sich ein Blatt oder Knoten im Graphen ändert. Die Nachteile von CSG, wie z.B. der beschränkte Satz an Basisprimitiven und die vergleichsweise langsame Verarbeitung des Baumes, haben dazu geführt, daß CSG im Sinne der ursprünglichen Definition kaum mehr eingesetzt wird. Stattdessen werden Boolsche Operationen häufig auf Modellen in *boundary representation* (BRep) durchgeführt [121]. Auch die im Rahmen dieser Arbeit entwickelte erweiterte Datenstruktur, der HistoryGraph, macht sich diese Kombination zunutze und erweitert die Möglichkeiten des CSG-Baumes auf vielfältige Weise.

3.4.1 Der HistoryGraph - eine erweiterte CSG-Datenstruktur

Der HistoryGraph stellt einen allgemeinen, azyklischen Graphen dar und unterscheidet sich somit grundlegend von einem CSG-Baum. Knoten und Blätter im History-Graphen enthalten gleichartige Informationen. Die in jedem Knoten und Blatt gehaltene Information unterscheidet sich von der in einem CSG-Baum gehaltenen Information. Die toplogischen und semantischen Unterschiede machen die gesteigerte Flexibilität des History-Graphen gegenüber dem CSG-Baum aus und ermöglichen u.a. die Erstellung von Versionen und Varianten. Darüber hinaus bildet der HistoryGraph eine Basis, um CAD-Modelle durch die Änderung ihrer Primitive zu modifizieren.

Ein Graph $G = (V, E)$ sei ein mit einem Bauteil korrespondierender HistoryGraph. Dann stellt die Menge E der Kanten die Abhängigkeiten der Teilmodelle dar, die das Bauteil bilden (siehe Abbildung 19). Die Menge V der Knoten enthält die Menge der Blätter. Die Blätter des History-Graphen repräsentieren die Primitive, aus denen das Modell gebildet wird. Zu den Primitiven gehören nicht nur klassische CSG-Grundkörper, sondern auch Linien, Linienzüge, Flächen, etc. Ein Knoten kennt die Operation aus der er hervorgegangen ist sowie deren Parameter. Operationen können beliebigen Typs sein; also nicht nur Boolsche Operationen wie bei CSG. Knoten verweisen auf die sie beeinflussenden Objekte (Knoten) sowie auf die, die durch sie beeinflußt werden. Schließlich enthält jeder Knoten drei Transformationszustände, nämlich die zum Zeitpunkt der Objekterzeugung gültige Transformation, die aktuelle Transformation und die vorherige Transformation. Dabei ist T_b die *Basis*-Transformation, die zur Zeit der Erzeugung gültig ist, T_c die *Current*-Transformation, die momentan gültig ist sowie T_f die *Former*-Transformation, die Transformation, die der aktuellen vorausging und für einen Undo/Redo-Mechanismus benötigt wird.

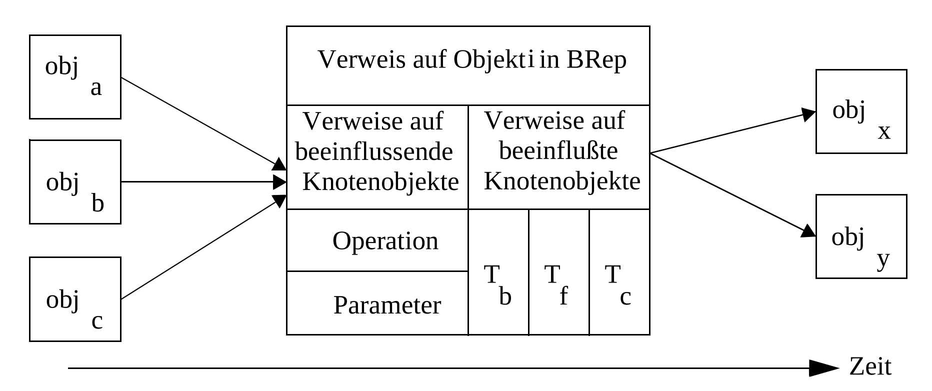


Abbildung 19: Chronologie des Konstruktionsprozesses, abgebildet im HistoryGraphen

Ein Knoten stellt somit ein Zwischenergebnis im Konstruktionsprozeß dar, eine Wurzel hingegen ein (bis auf weiteres) fertiggestelltes Modell. Die Zwischenergebnisse werden zur schnelleren Manipulation sowie zur Erzeugung von Versionen und Varianten permanent gehalten.

Der HistoryGraph spiegelt die chronologische Erzeugung eines Modells wider. Wird ein Objekt geändert, so ermöglicht die Graphenstruktur eine Vorwärtspropagierung dieser Änderung durch

den HistoryGraphen. Dabei werden alle vom geänderten Objekt abhängigen Objekte aktualisiert. Als Veränderung gelten sowohl Transformationen als auch Parameteränderungen von Objekten (Primitiven) oder Operationen. Prinzipiell läßt der HistoryGraph es zu, den Objekttyp innerhalb eines Knotens zu verändern oder auch die Operation gegen eine andere zu ersetzen; zu beachten ist, daß solche Ersetzungen nicht immer sinnvoll sind.

Anhand einer Transformation soll die automatische Propagierung von Änderungen im History-Graphen verdeutlicht werden; die Transformation, die ein abhängiges Objekt seit seiner Erzeugung erfahren hat, bleibt dabei erhalten:

> Zwei Objekte A und B seien erzeugt worden, so besitzen beide eine Basis-Transformation T_b^A bzw. T_b^B, die sich von der Indentität (id) unterscheiden, somit sind auch die aktuellen Transformation T_c^A und T_c^B von id verschieden. Die beiden Objekte werden einer Operation op unterzogen und es entsteht Objekt C mit einer Basistransformation T_b^C (siehe Abbildung 20a). Anschließend wird Objekt C vom Benutzer transformiert, so daß sich seine aktuelle Transformation T_c^C ändert (siehe Abbildung 20b). Nun wird Objekt B transformiert; damit ändert sich T_c^B . Der HistoryManager sorgt nach einer solchen Änderung dafür, daß alle von Objekt B abhängigen Objekte aktualisiert werden. Dies geschieht durch Re-Evaluierung der in den abhängigen Knoten gespeicherten Operationen und kann Auswirkungen auf deren Basistransformationen haben. Bevor Objekt C aus Objekt A und dem geänderten Objekt B neu bestimmt wird, wird die Differenztransformation berechnet, die es seit seiner Erzeugung erfahren hat - in diesem Beispiel T_1. Nun kann Objekt C aktualisiert werden und erhält damit eine geänderte Basistransformation. Abschließend wird die neue T_c^C aus der geänderten Basistransformation und der zuvor ermittelten Differenztransformation berechnet (siehe Abbildung 20c).

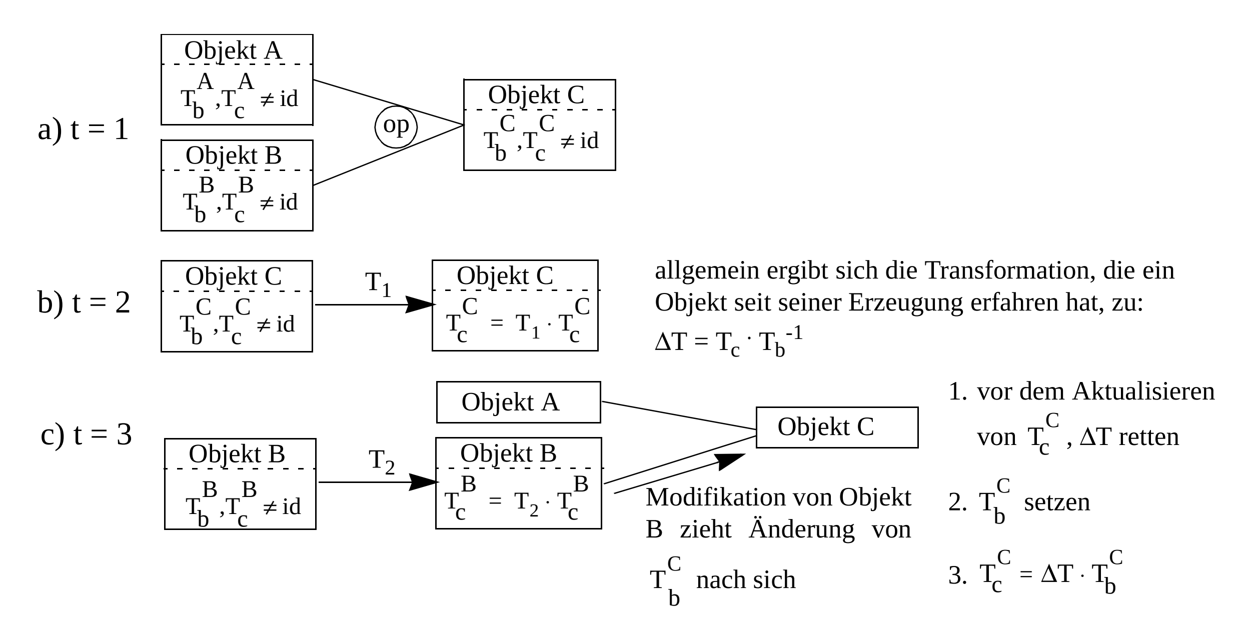


Abbildung 20: Propagierung von Änderungen durch den Historie-Graphen

Das anhand eines minimalen Beispiels dargestellte Verfahren läuft in komplexeren History-Graphen solange weiter, bis alle direkt oder indirekt abhängigen Objekte aktualisiert sind und propagiert somit Veränderungen, die der Benutzer in der Historie durchgeführt hat, automatisch ’vorwärts’. Dabei bleiben die Transformationen in den abhängigen Knoten erhalten.

Die Flexibilität des HistoryGraphen beliebige Operationen abbilden zu können, ermöglicht die Unterstützung von Varianten- und Versionenbildung. So wurden beispielhaft Kopier- und Instantiierungsoperationen implementiert. Letztere folgen den Ideen objekt-orientierter Ansätze und ermöglichen es, von einem Modell beliebig viele Instanzen zu bilden. Das Modell wird damit zum Master. Die Instanzen können unabhängig voneinander modifiziert werden. Ändert man den Master, so wirkt sich dies auf alle Instanzen aus.

Die beispielhaft implementierten Kopieroperationen vervielfältigen Modelle nicht nur, sondern sie können wahlweise dafür sorgen, daß die Kopie die Historie des Originals oder auch nur dessen Abhängigkeiten erbt. Die Kombination mit Transformationen ermöglicht die Bildung von Mustern (pattern) und Mustern von Mustern [156]. Der HistoryGraph ist auch die Basis einiger Interaktionsfunktionalitäten, die es erlauben, komplexe Modelle zu modifizieren oder Primitive auf einfachen Buttondruck zu vervielfältigen und automatisch in die Historie des entsprechenden komplexen Objektes einzufügen (vgl. *history pick* und *history copy pick* in Kapitel 4.2.7).

3.5 Der NetzwerkManager

Benutzer-zentrierte Ansätze fordern, den Kommunikationsbedarf des Benutzers bei der Ausführung seiner Arbeitsaufgabe durch geeignete Mechanismen zu unterstützen. Kommunikations- und Kooperationswerkzeuge ermöglichen heute schon kooperatives Arbeiten mit CAD-Systemen (siehe Kapitel 2). Ein benutzer-zentriertes Modelliersystem stellt allerdings erhöhte Anforderungen an den Kooperations- und Kommunikationsmechanismus:

- Es muß möglich sein, zwischen verschiedenen Ausprägungen des Systems zu kooperieren.
- Die direkt-manipulativen Prozesse der Partner sollen quasi in Echtzeit verfolgt werden können.
- Modelle sollen simultan kreiert und modifiziert werden können.
- Objekte sollen modifiziert werden können, solange kein anderer Partner daran arbeitet.
- Es soll erkennbar sein, wo die Partner im virtuellen, verteilten 3D-Konstruktionsraum arbeiten und von wo sie das Modell betrachten.
- Es soll möglich sein, eine Video-Konferenz innerhalb des Konstruktionsraumes abzuhalten.
- Es soll möglich sein, auf schmalbandigen Netzen (bis 2MBit/sec. Übertragungsrate) effizient zu kooperieren.

Der NetzwerkManager stellt innerhalb des Architekturkonzeptes für ein benutzer-zentriertes Modelliersystem das Modul dar, welches die genannten Anforderungen erfüllen soll. Ziel bei der Konzeption des NetzwerkManagers ist es, ein möglichst hohes Maß an Interaktivität und Telepräsenz bei möglichst effizientem Nachrichtenaustausch auf schmalbandigen Netzen zu ermöglichen.

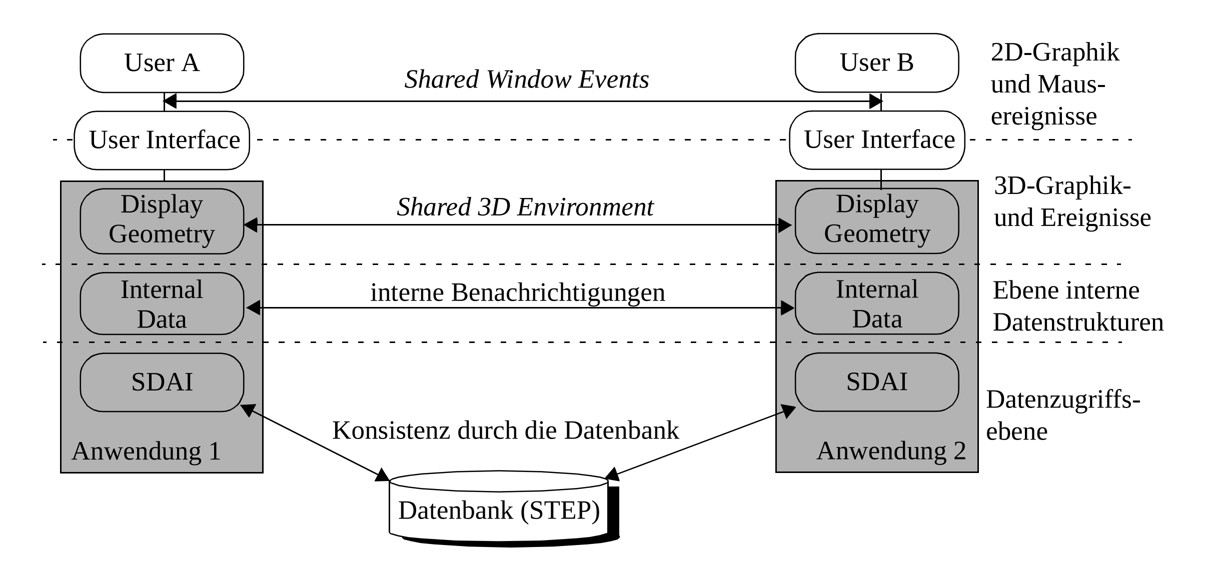


Abbildung 21: Die vier Ebenen der Systemkopplung [83]

Aus der Literatur [83] sind vier mögliche Ebenen zur Kopplung von Systemen bekannt, um kooperatives Verhalten zu erreichen (siehe Abbildung 21):

- Der Austausch von Bildinformation (Application Sharing und Shared Window Events).
- Der Austausch von 3D-Graphikdaten, z.B. die zum Zwecke der Darstellung triangulierte Repräsentation eines CAD-Modells (Shared 3D Environment).
- Der Austausch von Nachrichten auf der semantischen Ebene, d.h. Nachrichten, die im Rahmen des Anwendungskontextes Bedeutung haben, wie z.B. 'kreiere Zylinder' oder auch 'führe Boolsche Operation durch'.
- Die Kopplung über eine Datenbank.

Die vier Ansätze werden im folgenden im Hinblick auf die o.g. Anforderungen diskutiert.

Kopplung über eine Datenbank

Die Kopplung über eine Datenbank ist zur Unterstützung der verteilten synchronen Zusammenarbeit - dem Kooperieren über (große) Distanzen zur gleichen Zeit - weniger gut geeignet, da im Bereich der Produktentwicklung in einer Datenbank üblicherweise keine inkrementellen Änderungen, sondern Modelle in ihrer geometrischen Ausprägung abgelegt werden, z.B. im STEP-Format [6]. Ein Konvertieren in ein solches Datenformat auf der einen Seite und ein Dekonvertieren auf der anderen Seite schmälert die erzielbare Performanz und steht somit einer Echtzeit-Kopplung der Systeme entgegen.

Nachrichtenaustausch von Modellierkommandos

Der Austausch von Modellierkommandos in Form von Nachrichten ist prinzipiell geeignet, um kooperatives Modellieren zu unterstützen [48]. Die bekannten Ansätze übertragen allerdings nur das Ergebnis eines modell-erzeugenden oder -modifizierenden Dialogs. Die direkt-manipulativen Prozesse der Partner gehen dabei verloren. In einem benutzer-zentrierten System ist diesen interaktiven Prozessen mit geeigneten Mechanismen in besonderem Maße Rechnung zu tragen.

Austausch von 3D-Graphikdaten (Shared 3D Environment)

Der Shared 3D Environment-Ansatz ist zwar prinzipiell dazu geeignet, heterogene 3D-Graphikanwendungen zum kooperativen Arbeiten zu integrieren [83], allerdings reicht er alleine nicht aus, um kooperatives Modellieren zu ermöglichen. Beim Shared 3D Environment-Ansatz wird ein spezieller Knoten - der *networked node* - zur Verteilung szenengraph-basierter 3D-Graphikanwendungen benutzt. Dieser Knoten sorgt dafür, daß der lokale Szenengraph[2] sowie Änderungen am lokalen Szenengraph an die entfernten Anwendungen übertragen werden. Damit werden die 3D-Graphikdaten aller Partner an jedem Standort sichtbar. Weiterhin nimmt er die Ereignisse auf dem Szenengraph entgegen und stellt sie der Applikation zu, die den betroffenen Teil-Szenengraph beigetragen hat (siehe Abbildung 22). So können alle Partner mit allen Graphikobjekten interagieren.

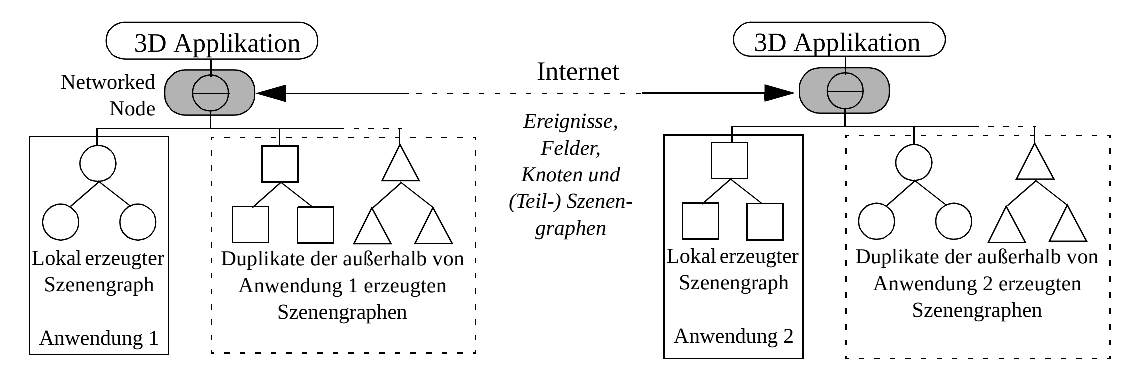


Abbildung 22: Prinzip des Austausches von 3D-Graphikdaten zur verteilten Visualisierung [149]

Der Networked Node arbeitet im Detail wie folgt:

Tabelle 2: Änderungsverwaltung im Networked Node [122]

Vorgang	Aktionen des Networked Node	gesendeter Nachrichteninhalt
erstmalige Knotenmodifikation durch Feldänderung	Speichern des Pfades zum geänderten Knoten	alle Felder des Knotens plus gespeicherter Pfad
erstmalige Knotenmodifikation ohne Feldänderung	Speichern des Pfades zum geänderten Knoten	Teilgraph plus gespeicherter Pfad
weitere Knotenänderung innerhalb des aktuellen Zeitintervalls	Suchen des letzten gemeinsamen Knotens des aktuellen Änderungspfades und dem gespeicherten Änderungspfad der vorherigen Änderung Speichern des Pfades zum gefundenen Knoten	Teilgraph plus gespeicherter Pfad
beim Senden am Ende des Zeitintervalls	falls der gespeicherte Änderungspfad nur den lokalen Wurzelknoten beinhaltet	vollständiger lokaler Szenengraph
keine Änderung im aktuellen Zeitintervall	-	-

Obwohl die Kooperationspartner beim Shared 3D Environment-Ansatz alle Objekte sehen und auch mit allen Objekten interagieren können, sind die Möglichkeiten, auf den Objekten Opera-

2. Ein Szenengraph ist eine mögliche Repräsentationsform einer Part-Assembly-Struktur in einem Graphiksystem.

tionen auszuführen beschränkt. Möchte man diesen Ansatz zum kooperatien Modellieren einsetzen, so steht man vor dem Problem, daß die entfernten Instanzen von einem Modell nur dessen graphische Repräsentation kennen. Modelle eines entfernten Partners können damit nicht für Modellierungsoperationen auf dem lokalen Rechner herangezogen werden. Das bedeutet, daß jeder Partner in einer kooperativen Sitzung nur mit den vom ihm erzeugten Modellen arbeiten kann; allerdings wird das Ergebnis einer lokalen Operation den Partern dargestellt.

Austausch von 2D-Bildinformation

Der Austausch von Pixelbildern, das Application Sharing, ist auf 2D-Bildinformation und 2D-Ereignisse limitiert. Application Sharing koppelt nicht zwei Instanzen eines Systems, sondern stellt bei den Kooperationspartnern nur ein Abbild der Ausgabe dar, die auf dem Rechner erzeugt wird, auf dem das System tatsächlich läuft. Die Übertragung von Pixelbildern beansprucht relativ viel Netzwerkbandbreite. Um dieses Problem zu mildern, versuchen Entwickler von Application Sharing-Systemen neuerdings auf den Strom von Graphikbefehlen, z.B. OpenGL-Befehle, zurückzugreifen und diesen zu optimieren [133].

Im Rahmen eines diese Arbeit begleitenden F&E-Projektes zur Einführung von CSCW-Werkzeugen in den Konstruktionsprozeß [3], [171] wurde die von einem Application Sharing Tool erzeugte Netzlast ermittelt. Zum Vergleich wurde eine nachrichten-basierte CSCW-Erweiterung des Systems FeatureM [153] realisiert. Konstrukteure wurden hinsichtlich der Akzeptanz beider CSCW-Lösungen befragt. Obwohl eine Leitung mit 2Mbit/sec zur Verfügung stand, waren die Konstrukteure mit dem Antwortzeitverhalten des Application Sharing Tools unzufrieden und bevorzugten ein einfaches White Board-System, das die herkömmliche Arbeitsweise der Problemdiskussion mittels Fax und Telefon zu ersetzen in der Lage ist [170]. Andere Arbeiten [8] bestätigen die Aussage, daß Application Sharing über schmalbandige Netze, wie z.B. ISDN, ein unbefriedigendes Antwortzeitverhalten zeigt und mehr als 2MBit/sec. benötigt werden, um verzögerungsfrei arbeiten zu können. Beim Application Sharing können die Partner nur abwechselnd arbeiten. Somit erfüllt Application Sharing die o.g. Anforderungen nicht.

3.5.1 Kombinierter Nachrichtenaustausch auf zwei Ebenen

Um die genannten Anforderungen an die Kommunikations- und Kooperationskomponente eines benutzer-zentrierten 3D-Modellierungssystems zu erfüllen, wird das Konzept eines kombinierten Nachrichtenaustauschs auf zwei Ebenen entwickelt. Der NetzwerkManager kombiniert den Austausch von semantischen Nachrichten mit dem Austausch von 3D-Graphikdaten und bietet damit simultanes, direkt-manipulatives und kooperatives Modellieren. Um dies auf einem konsistenten Weg zu gewährleisten, besitzt der NetzwerkManager zwei Komponenten. Eine kommuniziert mit dem ObjektManager, die andere mit dem GraphikManager (siehe Abbildung 23). Aus dem ObjektManager bezieht der NetzwerkManager die ausgeführte Modellierungsoperationen und überträgt diese an die entfernten Instanzen, wo sie nachgeführt werden. Aus dem GraphikManager bezieht der NetzwerkManager alle direkt-manipulativen Aktionen des Benutzers sowie die

Position des Cursors und der Kamera und überträgt diese an die entfernten Instanzen. Die remote 3D cursor und die Avatare bieten ein hohes Maß an Telepräsenz.

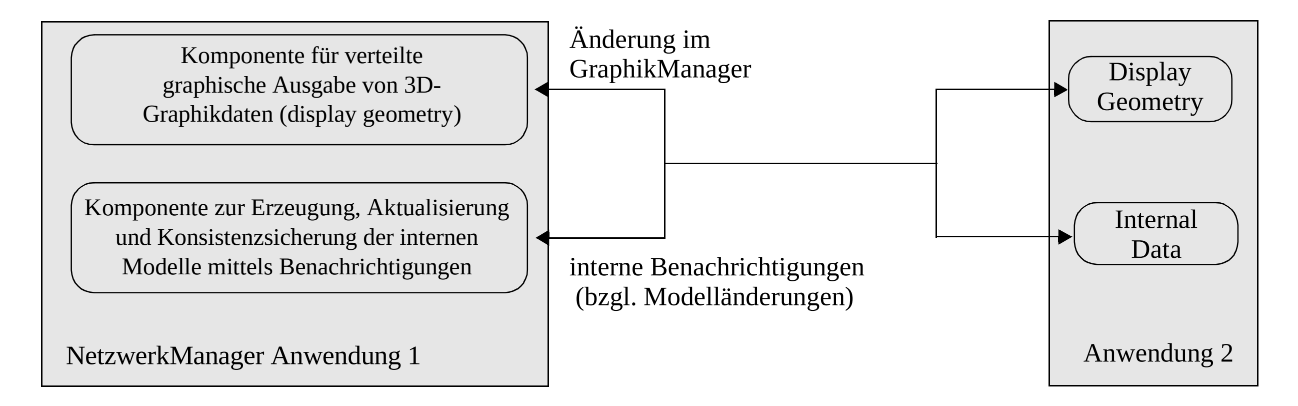


Abbildung 23: Kombination zweier Kommunikationsebenen durch den NetworkManager [122]

Nachrichten auf graphischer und Nachrichten auf interner Ebene laufen über einen Kommunikationskanal. Eine derartige Sequentialisierung der von einer Instanz ausgesendeten Nachrichten ist notwendig, um deren Reihenfolge zu wahren. Simultanes Arbeiten wird durch die ereignisorientierte Bearbeitung eintreffender Nachrichten sichergestellt, so kann z.B. ein gerade durch den entfernten Partner direkt-manipulativ erzeugtes Objekt in die Szene aufgenommen und dargestellt werden, auch wenn der lokale Bearbeiter gerade ein Modell ändert. Der Versuch, zeitgleich auf ein und demselben Objekt arbeiten zu wollen, muß hingegen von dem System erkannt und unterbunden werden, um widersprüchliche Modifikationen eines Objektes durch verschiedene Benutzer und somit Inkonsistenzen zu vermeiden. Konzepte zur Konfliktauflösung im Falle konkurrierender Ressourcenanforderung können hier eingesetzt werden. Ein geeignetes Kommunikationssystem, auf dessen Basis der NetzwerkManager umgesetzt und das vorliegende Konzept evaluiert wurde, steht mit Request++ [4], [155] zur Verfügung.

Mit Hilfe der von Request++ zur Verfügung gestellten Edikte zur Ressourcenanforderung wird ein fließendes objekt-bezogenes Aktionsrecht realisiert, bei dem ein Objekt vor dem Zugriff durch andere Benutzer nur solange gesperrt ist, wie einer der Kooperationspartner in einer Sitzung dieses Objekt manipuliert. Nachdem die Manipulation abgeschlossen ist, steht das Objekt wieder allen Beteiligten zur Verfügung. Das objekt-bezogene, fließende Aktionsrecht zeigt, was technisch an Flexibilität in kooperativen Anwendungen machbar ist; tatsächlich kann die Granularität, auf der der Zugriff verwaltet wird, noch verfeinert werden. Bei der Einführung von CSCW-Technologien in die betriebliche Praxis sind sehr oft restriktivere Handhabungsweisen erforderlich, da das Eigentums- und Änderungsrecht einen der sensibelsten Punkte in der Produktentwicklung darstellt. Die 3D-CAD-Daten entwickeln sich zunehmend zum digitale Master eines in der Entwicklung befindlichen Produktes. Sie sind für ein Unternehmen von höchstem Wert, da Know-How und Innovation mehr oder weniger offensichtlich in ihnen enthalten ist. 3D-CAD-Daten neuester Entwicklungen werden - wenn überhaupt - nur mit Geheimhaltungsabkommen an Dritte weitergegeben und dürfen von diesen nur genutzt, nicht aber modifiziert werden. Fragen der Authentisierung von Produktmodellen beschäftigen die Forschung [9]. Einen anderen Ansatz das Know-How am Produkt zu schützen, stellt die funktionale Verfremdung unter weitestgehender Wahrung der geometrischen Gestalt mittels Wavelet-Transformationen dar [104]. So können mit 3D-Modellen von einem Auftrageber z.B. Einbauuntersuchungen durchgeführt werden, ohne daß dieser das in der exakten Geometrie steckende Know-how preisgibt. Eine Alternative zur Authentisierung besteht im 3-dimensionalen digitalen Wasserzeichen (digital watermarking in 3D), also

dem urheberrechtlichen Kennzeichnen von 3D-Daten, das inhärent mit diesen verbunden ist. Diese Forschungs- und Entwicklungstendenzen zeigen, daß es in Zukunft darauf ankommen wird, die technisch mögliche Flexibilität bei der Handhabung von Produktmodellen an die organisatorischen und rechtlichen Gegebenheiten von Unternehmen anzupassen. Workflow- und Produktatenmanagementsysteme [7] stellen unternehmensintern dazu notwendige Schritte dar.

3.5.2 Simultane, verteilte Objekterzeugung und -modifikation

Anhand der Objekterzeugung und -modifikation soll nun gezeigt werden, wie der Shared 3D Environment-Ansatz und der Nachrichtenaustausch auf semantischer Ebene ineinandergreifen, um simultanes, direkt-manipulatives und kooperatives Modellieren zu erreichen.

Während der Erzeugung eines Objektes durch einen Benutzer wird die graphische Repräsentation des Objektes, d.h. der dem Objekt zugeordnete Szenengraph, überwacht. Veränderungen innerhalb dieser Teilszene werden an die beteiligten Partnerinstanzen übermittelt und dort in den Szenengraphen eingebaut (siehe Abbildung 24). Somit wird der direkt-manipulative Entstehungsprozeß eines Objektes durch sukzessive Propagierung der Änderung allen Partnern dargestellt.

Simultan zur lokalen Objekterzeugung können von den anderen Partnern stammende Teilszenengraphen in die lokale Instanz eingebracht werden, was seinerseits dazu führt, daß die Änderungen der entfernten Partner lokal sichtbar werden. D.h. die lokale Ereignisverarbeitung, das Senden von Nachrichten und das Empfangen von Nachrichten laufen quasi gleichzeitig ab. Somit lassen sich aus Sicht jedes Benutzers mehrere Objekte von verschiedenen Partnern simultan erzeugen und derer Erzeugungsvorgänge in Echtzeit beobachten.

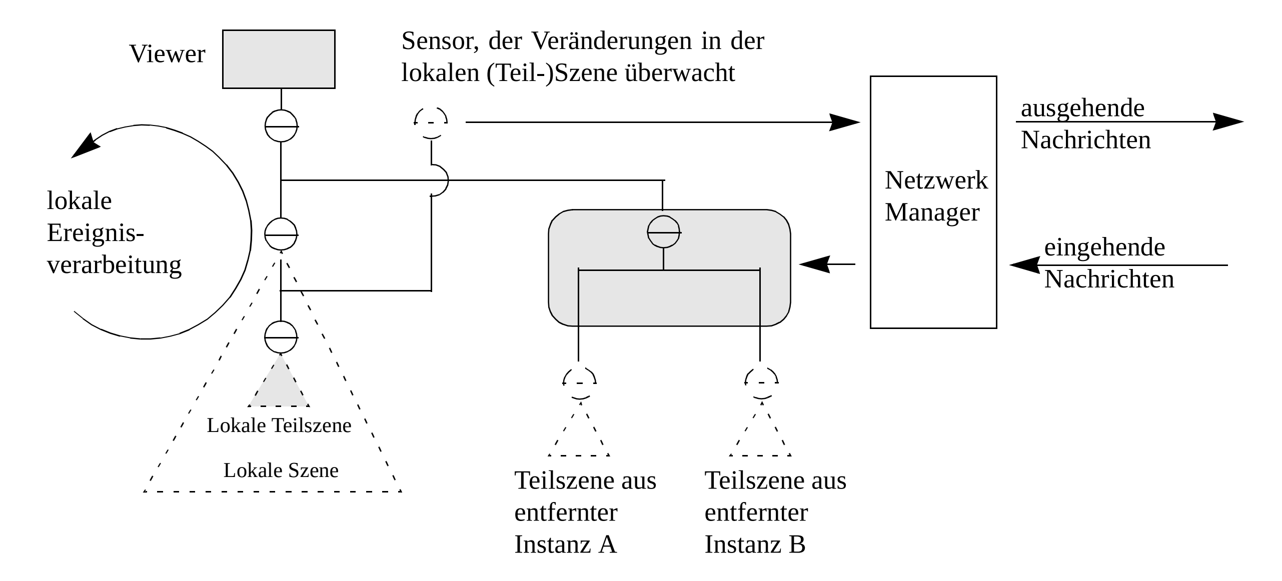


Abbildung 24: Versendung der Teilszene während der interaktiven Objekterzeugung

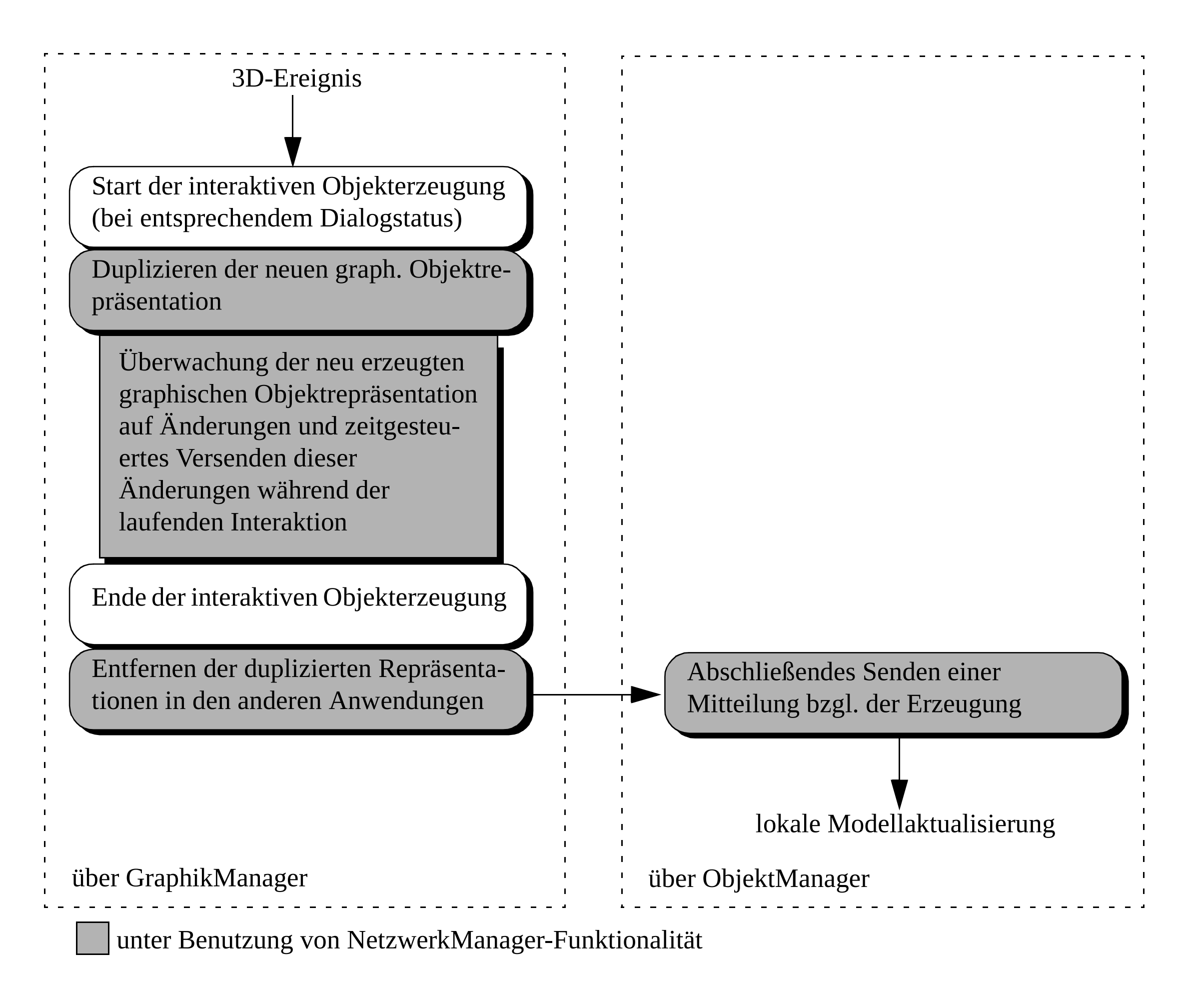


Abbildung 25: Verteilte Objekterzeugung [122]

Während der interaktiven Erzeugung werden nur graphische 3D-Daten ausgetauscht. Mit Abschluß einer Objekterzeugung wird eine Nachricht auf semantischer Ebene generiert, z.B. create<ObjectType>Msg [122], die bei den entfernten Instanzen zu einer Objektinstantiierung mit allen seinen Repräsentationen und semantischen Informationen führt (siehe Abbildung 25). Bei Abbruch der interaktiven Objekterzeugung wird ebenfalls eine Nachricht an die beteiligten Sessioninstanzen geschickt, die zur Folge hat, daß die graphische Repräsentation aus dem Szenengraph entfernt wird.

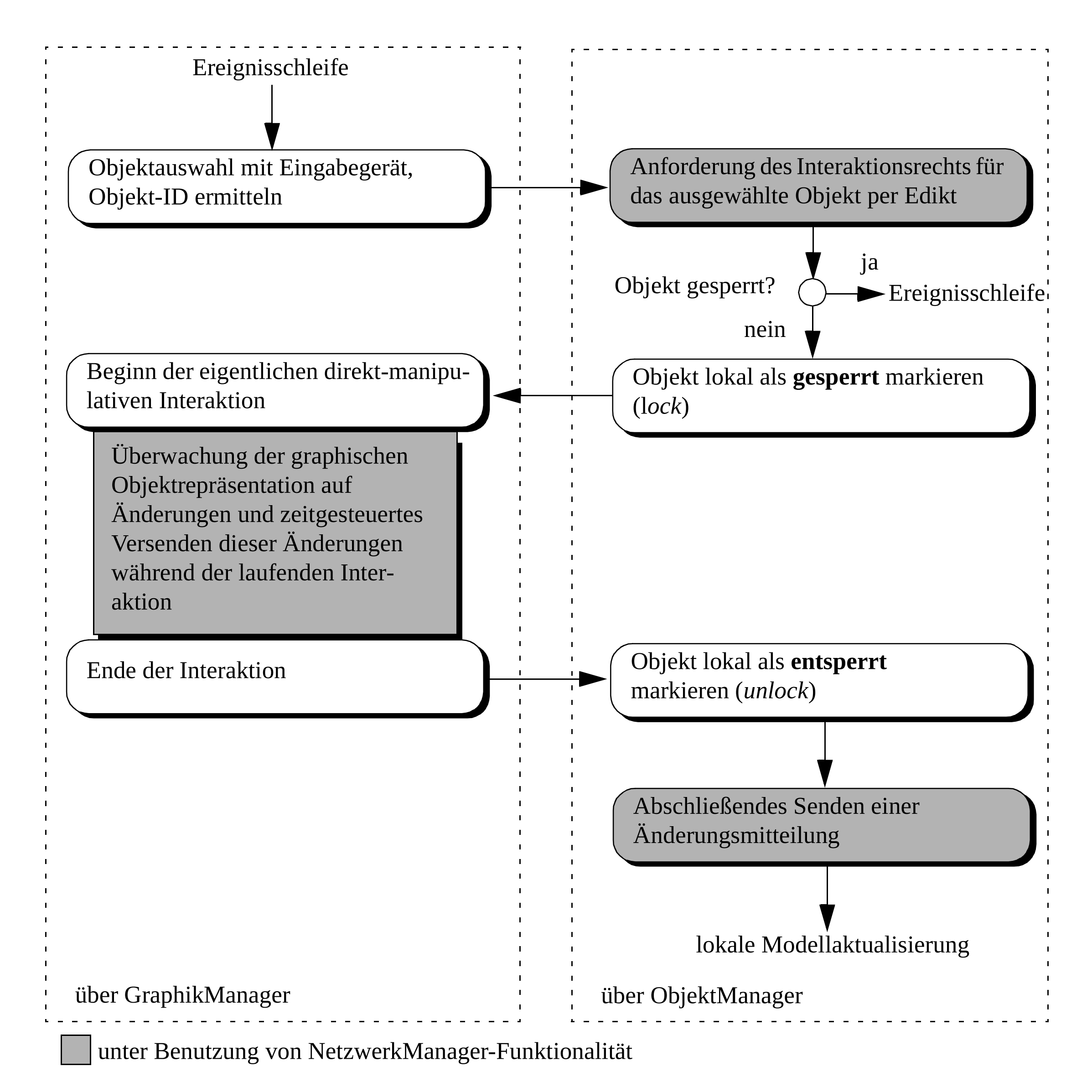


Abbildung 26: Verteilte Objektmanipulation [122]

Die verteilte Modellmodifikation läuft ähnlich ab, allerdings ist hier die Aktionsrechtanforderung vorgeschaltet, um konkurrierende Zugriffe zu vermeiden. Zunächst erfolgt die Ressourcenanforderung mittels Edikt. Wird das Objekt von einem anderen Benutzer manipuliert, so weist das System den Wunsch des Benutzers zurück, mit dem betreffenden Objekt zu arbeiten. Wird das Objekt nicht anderweitig manipuliert, so erhält der Benutzer das Aktionsrecht für dieses Objekt und sperrt es damit vorübergehend für den Zugriff durch andere. Der Benutzer kann nun das Objekt manipulieren, z.B. seine Parameter oder Transformation ändern. Diese Änderungen geschehen direkt-manipulativ im GraphikManager. Zur Überwachung der Änderungen wird, wie bei der Objekterzeugung, der entsprechende Teilszenengraph beobachtet. Änderungen werden als Nachrichten auf graphischer Ebene an die Sessioninstanzen übertragen (siehe Abbildung 26). Bei Beendigung der direkten Manipulation wird die Sperrung für das bearbeitete Objekt aufgehoben und eine Änderungsnachricht auf semantischer Ebene zur Konsistenzsicherung durch den Objekt-Manager geschickt. Da das Locking objekt-bezogen arbeitet, können von der Teilnehmern einer kooperativen Sitzung verschiedene Objekte zur gleichen Zeit modifiziert werden, wobei die Modifikation bei den anderen Partnern quasi in Echtzeit nachvollzogen werden kann.

Modellierungsoperationen, wie z.B. Boolsche Verschneidungen und Sweeping, werden durch Nachrichten auf semantischer Ebene propagiert. Dazu erfolgt zunächst die Selektion der beteiligten Objekte. Die Selektion wirkt sperrend auf Objekte, um sie vor konkurrierendem Zugriff durch andere Partner zu schützen. Danach initiiert der Benutzer lokal die durchzuführende Operation. Die Operation wird kodiert und zusammen mit den involvierten Objekten verschickt. Abschließend wird die Operation sowohl lokal als auch bei den entfernten Instanzen ausgeführt und die gesperrten Objekte wieder freigegeben.

Eines der Ziele des NetzwerkManagers ist, auf verhältnismäßig schmalbandigen Netzen performant kooperieren zu können. 3D-Graphikdaten sind nicht notwendigerweise kompakt und damit netzlastschonend. Aus diesem Grund wurden folgende Verfahren zur Minimierung der Netzwerklast bei der Realisierung des NetzwerkManagers eingeführt:

- Intelligentes Versenden von Teilszenen

  Beim Überwachen von Teilszenengraphen wird erkannt, ob die Veränderung nur einen Knoten im Szenengraph oder nur ein Feld eines Knotens betrifft. Es wird immer die kleinstmögliche Einheit übertragen.
- Zeitgesteuertes Versenden von Teilszenen

  Direkte Manipulationen lassen sich lokal - je nach Eingabegerät und Bildwiederholfrequenz - mit mehr als 50 Hz darstellen. Zwischen zwei aufeinanderfolgenden Ereignissen finden oft nur kleinste Änderungen statt. Deren Übertragung stellt eine unnötige Netzlast dar. Eine Aktualisierung der entfernten Instanzen mit 10 bis 20 Hz hat sich als völlig ausreichend herausgestellt und hilft die Netzlast zu minimieren. Nach Beendigung einer Interaktion wird unabängig vom Zeitraster durch Übertragung des erreichten Endzustands für Konsistenz gesorgt.

Geometriekompression könnte die Netzlast weiter verringern. Davon wurde bei der Implementierung des NetzwerkManagers allerdings abgesehen, da sie eine Lastverteilung zu ungunsten der CPU darstellt. Die zu erwartenden Ergebnisse würden mit der Rechengeschwindigkeit der lokalen sowie der entfernten Hardware und der Netzbandbreite variieren. Eine Balancierung der Last, die diese Parameter einbezieht, wäre nötig, die ihrerseits einen Aufwand darstellt.

3.5.3 Telepräsenz im verteilten, virtuellen 3D-Konstruktionsraum

Telepräsenz wird in herkömmlichen CSCW-Lösungen, wie Application Sharing und Video-Konferenzsystemen, durch Darstellung der Mauszeiger der entfernten Benutzer und deren Videobild erreicht. Das Videobild wird in einem eigenen Fenster auf dem Bildschirm dargestellt. Obwohl das Videobild für die Übermittlung von Gestik und Mimik an entfernte Benutzer als sehr wichtig eingeschätzt wird [157], wird es während einer Kooperation mittels Application Sharing von den Beteiligten kaum beachtet, da es außerhalb der Applikation dargestellt wird und ein Hin- und Herschauen des Benutzers erfordert. Der Benutzer wird dadurch von der eigentlichen Arbeitsaufgabe bzw. vom Diskussionsobjekt abgelenkt.

Um ein höheres Maß an Telepräsenz zu erreichen, wurde die Möglichkeit zur Integration einer Videokonferenz direkt in den verteilten virtuellen Konstruktionsraum geschaffen [152]. Das Videobild eines entfernten Benutzers wird dabei als animierte Textur auf einen ’virtuellen Bildschirm’ innerhalb der Szene gelegt, der die Position und Orientierung des Partners visualisiert (siehe Abbildung 27). Der lokale Benutzer kann somit seinem Partner virtuell gegenübertreten, mit ihm reden und sein Videobild sehen, während beide simultan in dem verteilten virtuellen Konstruktionsraum arbeiten. Die Visualisierung der Partner in Form von Avataren bietet dem

lokalen Benutzer die Möglichkeit zu erkennen, welchen Interessensbereich ein bestimmter Partner hat. Durch Anklicken des Avatars kann eine Synchronisierung der Sichten der Partner auf die Szene erfolgen, wobei die Partner jederzeit die Möglichkeit haben, unabhängig ihre Position zu wählen. Die integrierte Videokonferenz erfordert Hardware-unterstütztes Texturieren und steigert - wie jede Videokonferenz - die Netzwerklast.

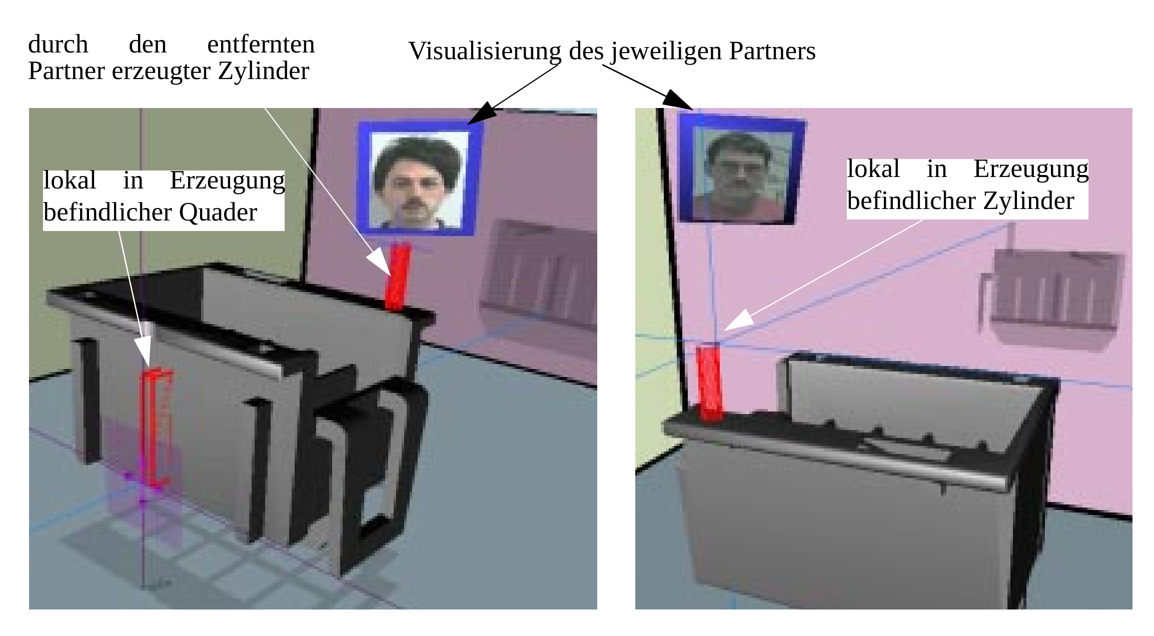


Abbildung 27: Zwei Screenshots einer kooperativen Modellierungssitzung; links die Sicht des einen Partners, rechts die des anderen. Die als Drahtgittermodell dargestellten Primitive (Zylinder und Quader) werden simultan von den beiden Benutzern erstellt.

Können die Sichten auf das Modell von den Partnern unabhängig gewählt werden, so wird die Position eines remote 2D cursors, wie er von Application Sharing-Systemen dargestellt wird, um die Aktion des entfernten Benutzers dem lokalen Betrachter zu vermitteln, bedeutungslos. Stattdessen muß eine szenen-inhärente 3-dimensionale Rückkopplung erfolgen. Der NetzwerkManager tut dies in Verbindung mit dem GraphikManager durch die Visualisierung eines remote 3D cursors, der den Bewegungen des Cursors des entfernten Benutzers folgt und als Fadenkreuz mit beschränkten Ausmaßen dargestellt wird, während der lokale 3D Cursor den gesamten Konstruktionsraum durchspannt.

Abschließend bleibt zu erwähnen, daß der NetzwerkManager nicht nur das Arbeiten zwischen zwei Benutzern unterstützt, sondern prinzipiell beliebig viele Partner zuläßt. Die visuelle Zuordnung der remote 3D cursor zu den Avataren wird mittels einer eindeutigen Farbkodierung erleichtert.

Durch die Echtzeitvisualisierung der Direkt-Manipulationen der Partner, die Möglichkeit der in die Szene integrierten Video-Konferenz sowie Avatare und remote 3D cursor bietet der Netzwerk-Manager ein zuvor unerreichtes Maß an Telepräsenz für kooperative Modellierungsanwendungen. Durch die Unabhängigkeit des NetzwerkManagers von der konkreten Hardwareumgebung ist es möglich, unterschiedlich konfigurierte Systeminstanzen zu verbinden und somit Kooperationsmöglichkeiten zwischen herkömmlichen und (semi-) immersiven Arbeitsplätzen zu realisieren (siehe Abbildung 28).

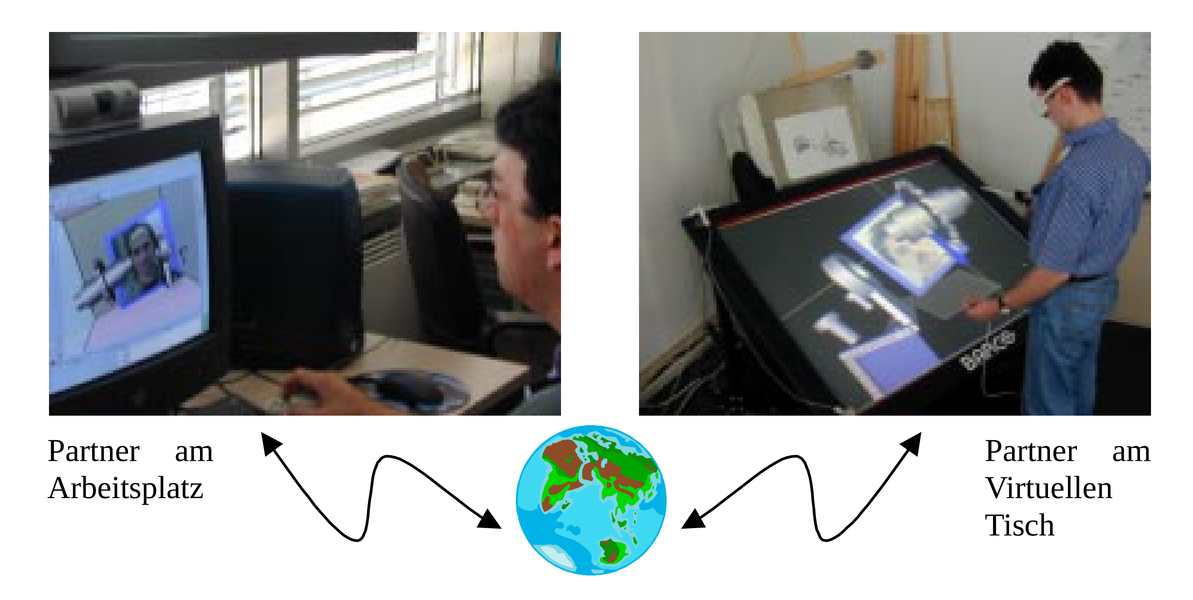


Abbildung 28: Kooperatives Modellieren in einer hybriden Umgebung

3.5.4 Netzlast nachrichten-basierter Ansätze

Wie bereits erwähnt, ist sinnvolles Arbeiten mit Application Sharing von CAD-Systemen erst ab Netzwerkbandbreiten von mindestens 2MBit/sec. möglich. Diese Bandbreiten sind im Bereich lokaler Netze üblich und kostengünstig zu realisieren, wenn auch das stark verbreitete betriebsinterne 10 oder gar 100 MBit/sec. Ethernet von vielen Benutzern gleichzeitig belastet wird und somit die jedem Benutzer zur Verfügung stehende Netzkapazität stark variiert.

Im Bereich der Weitverkehrsnetze (wide area networks WANs) sieht das Verhältnis von Netzkapazität und -kosten weitaus schlechter aus. Hier sind die Kosten für Netzkapazitäten von 2 MBit/sec. und mehr für klein- und mittelständische Unternehmen (KMUs) wirtschaftlich nicht tragbar. Gerade diese Betriebe unterhalten vielfältige Beziehungen zu ihren räumlich oftmals entfernten Auftraggebern oder verfügen selbst über Zweigstellen, mit denen es zu kommunizieren und kooperieren gilt [170], um einen reibungslosen, effizienten und kostengünstigen Entwicklungs- und Fertigungsprozeß zur Erstellung qualitativ hochwertiger Produkte zu erreichen.

Für ein solches Unternehmen wurde - in einem die vorliegende Arbeit begleitenden Forschungs- und Entwicklungsprojekt - dem nachrichten-basierten Konzept des NetzwerkManagers folgend, eine CSCW-Lösung für ein kommerzielles CAD-System geschaffen. Die Umsetzung basiert auf CORBA [108] und verzichtet auf die Übertragung direkt-manipulativer Prozesse und eine integrierte Videokonferenz. In einer kooperativen Sitzung wurde über 10 Minuten hinweg die erzeugte Netzlast gemessen und der eines Application Sharing-Systems gegenübergestellt, mit dem dieselbe kooperative Sitzung durchgeführt wurde. Dafür stand exklusiv für den Test eine 2 MBit/sec. Verbindung zwischen zwei ca. 200 km voneinander entfernten Werken zur Verfügung. Der nachrichten-basierte Ansatz erzeugte eine gemittelte Netzlast von 4.8 KBit/sec. Das Application Sharing Tool induzierte mit ca. 250 KBit/sec. eine um den Faktor 50 höhere Netzlast. Einer Befragung der beteiligten Konstrukteure zufolge, wurde das Antwortzeitverhalten der nachrichten-basierten Lösung trotz der geringen Menge an übertragenen Daten als besser empfunden. Die Konstrukteure beurteilten den nachrichten-basierten Ansatz für kooperatives Modellieren als gut geeignet; Application Sharing auf einem 2Mbit/sec.-Netz als nicht akzeptabel.

Die geringeren Netzkosten und die höhere Arbeitsgeschwindigkeit eines solchen Systems im laufenden Betrieb müssen dem erhöhten Implementierungsaufwand gegenübergestellt werden. Selbst bei einer kostenintensiven Individuallösung kommt man auf Armortisierungszeiten im Bereich um ein Jahr [157]. Würden die CAD-Anbieter generell eine nachrichten-basierte Lösung realisieren und anbieten und deren Kosten auf die Lizenzgebühren umlegen, so wären die Zusatzkosten für den CAD-Anwender durchaus mit den Investitionen in ein Application Sharing Tool vergleichbar und somit kooperatives Modellieren auf schmalbandigen Netzen für viele potentielle Nutzer interessant und erschwinglich.

Die Umsetzung des NetzwerkManagers innerhalb des Systems ARCADE zeigt, daß simultanes kooperatives Modellieren bei Übertragung der direkt-manipulativen Prozesse mit Netzbandbreiten sinnvoll machbar ist, die unter denen für Application Sharing benötigten Netzbandbreiten liegen. Über eine transatlantische ATM-Verbindung, die streckenweise 512 Kbit/sec. bzw. 1.5 MBit/sec. Netzkapazität zur Verfügung stellte, konnte gezeigt werden, daß die Partner einer kooperativen ARCADE-Sitzung das Gefühl haben, verzögerungsfrei miteinander arbeiten und die Ergebnisse der anderen Partner sehen zu können. Das zeigt, daß mit dem durch den NetzwerkManager eingeführten Konzept des Nachrichtenaustausches auf zwei Ebenen, verteiltes, simultanes und direkt-manipulatives Modellieren bei gesteigerter Telepräsenz auf schmaleren Netzen als Application-Sharing möglich ist, wobei letzteres die genannten fortschrittlichen Merkmale nicht aufweist.

3.6 Zusammenfassung

In diesem Kapitel wurde ein Architekturkonzept für benutzer-zentrierte 3D-Modellierungssysteme eingeführt, das den z.T. differgenten Anforderungen von CAD, CSCW und VR Rechnung trägt. Mit dem entwickelten Architekturkonzept verschwimmen die Grenzen zwischen kooperativem CAD und verteiltem VR; jeder Benutzer wählt die Interaktions- und Präsentationsform, die seiner Aufgabe am nächsten kommt.

Zwei Komponenten sind zur Erfüllung der Anforderungen benutzer-zentrierter Modellierungssysteme in diesem Architekturkonzept von besonderer Bedeutung: der GraphikManager und der NetzwerkManager. Dem HistoryManager kommt nicht die zentrale Rolle des Graphik- bzw. NetzwerkManagers zu, aber auch er dient dazu, den Benutzer bei der effizienten Bearbeitung seiner Aufgaben zu unterstützen.

Der GraphikManager, bestehend aus Interaktions-, Präsentations- und Direkte-Manipulationskomponente, ist mit Modellierungsfunktionalität angereichert und verfügt über die Möglichkeit, VR-Gerätetechnologie anzukoppeln. Mit dem Konzept des *user-in-the-loop* und schneller, auf Basis der graphischen Repräsentation der CAD-Modelle im GraphikManager arbeitender Methoden zur Modellierung wird die Anforderung nach Direkt-Manipulation erfüllt.

Eine CSCW-Komponente ist integraler Bestandteil des Konzeptes. Sie erlaubt durch kombinierten Nachrichtenaustausch auf der graphischen und semantischen Ebene simultanes, kooperatives Modellieren mit Echtzeit-Feedback der Prozesse der entfernten Benutzer. Die Möglichkeit, eine Videokonferenz direkt in die 3D-Szene zu integrieren, steigert die Telepräsenz nochmals. Der Kommunikations und Kooperationsmechanismus ist unabhängig von der Hardware-Konfiguration einer Instanz, so daß Kooperation zwischen einem Desktop- und einem (semi-)immersiven Arbeitsplatz möglich ist.

Der HistoryManager stellt die Komponente dar, die die Schnittstelle zwischen dem Benutzer und dem HistoryGraphen - einer erweiterten CSG-Datenstruktur - bildet und die Konstruktionshistorie

verwaltet. Der HistoryGraph ist eine Verallgemeinerung des CSG-Baumes. Er bietet vielfältige Möglichketen zur Erzeugung und Beeinflussung von Varianten und Versionen [156]. Einige der im nächsten Kapitel vorgestellten Interaktionstechniken für die 3D-Modellierung mit 3D-Eingabegeräten machen sich den HistoryGraphen zunutze.

4 3D-Interaktions- und Visualisierungstechniken

In diesem Kapitel werden die im Rahmen dieser Arbeit entwickelten 3D-Interaktionstechniken dargestellt. Der Schwerpunkt liegt hierbei auf der Erschließung von 3D-Eingabegeräten für den präzisen Konstruktionsprozeß durch geeignete 3D-Interaktionstechniken. 3D-Eingabegeräte weisen 3, 6 oder sogar mehr Freiheitsgrade auf. Einerseits erlaubt eine hohe Anzahl von Freiheitsgraden eine natürliche Interaktion mit virtuellen Objekten, andererseits wird in der Bauteilmodellierung oft nur eine reduzierte Anzahl von Freiheitsgraden benötigt. Die Kernfrage, die es zu klären gilt, lautet: Wie lassen sich die theoretischen Vorteile von 3D-Eingabegeräten unter Beachtung der motorischen Fähigkeiten des Menschen im 3D-Modellierungsprozeß vorteilhaft nutzen?

Zu diesem Zweck werden nicht nur neuartige 3D-Interaktionstechniken entwickelt, sondern auch unterstützende Visualisierungstechniken, da der visuellen Rückkopplung interaktiver Prozesse in 3D eine erhöhte Bedeutung zukommt. Nur durch entsprechende Visualisierung kann der virtuelle 3-dimensionale Raum, in dem interagiert wird, dem Betrachter auf einem Ausgabegerät herkömmlicher Bauweise[1] vermittelt werden.

4.1 3D-Visualisierungstechniken

Unter Visualisierung wird im allgemeinen das Sichtbarmachen von rechner-internen Informationen verstanden. Als Visualisierungstechnik bezeichnet man eine bestimmte Art und Weise, wie eine gegebene Menge von Daten dargestellt wird. Das Ziel von Visualisierungstechniken ist es, dem Betrachter das Erkennen in den Daten erhaltener Strukturen d.h. Informationen durch geeignete Darstellung zu vereinfachen. Seit kurzer Zeit wird auch der Begriff Perzeptionalisierung verwendet [130], der die Beachtung wahrnehmungspsychologischer Erkenntnisse stärker zum Ausdruck bringt.

Im Anwendungsfeld Modellierung stellen 3D-CAD-Modelle die zu visualisierenden Daten dar. Für den Betrachter relevante Informationen sind die Form der Objekte, deren Parameter, Abhängigkeiten zwischen Teilmodellen, deren relative Lage usw. Zwar wird die früher bei kommerziellen CAD-Systemen anzutreffende Drahtgitterdarstellung immer stärker von einer schattierten Darstellung verdrängt, die mehr Formhinweise bietet, allerdings bieten die CAD-Systeme keine expliziten Hinweise auf die räumliche Anordnung von Objekten. Um die Orientierung, Navigation und Positionierung in 3D zu erleichtern, sind neben geeigneten Interaktionstechniken auch unterstützende Visualisierungstechniken nötig, die dem Benutzer aufschlußreiche Form- auch Tiefenhinweise bieten. Das Erkennen räumlicher Beziehungen zwischen Objekten ist bei Verwendung

1. Unter Ausgabegeräten herkömmlicher Bauart werden hier alle - auch stereoskopische - Projektionsverfahren verstanden, da die heutige Displaytechnologie nur Pseudo-3D-Darstellungen in Echtzeit ermöglicht. Eine ’echte’ 3D-Darstellung wäre hingegen ein interaktives Hologramm.

von 3D-Eingabegeräten essentiell. Ohne eine Visualisierung räumlicher Informationen ist der Einsatz eines 3D-Eingabegerätes nur mit reduzierter Effizienz möglich. Bei der Entwicklung geeigneter Visualisierungstechniken sollen konform der Forderung benutzer-zentrierter Systeme die wahrnehmungspsychologischen Eigenschaften des menschlichen visuellen Systems beachtet werden.

Im folgenden wird diskutiert, welche der aus der Realität bekannten Effekte zur monokularen bzw. binokularen Tiefenwahrnehmung in einem Modellierungssystem unter dem Aspekt der Echtzeitfähigkeit umgesetzt werden können; die Forderung nach Echtzeitfähigkeit steht dabei der Forderung nach Realitätstreue entgegen. Anschließend wird deren Umsetzung beschrieben und es werden weiterführende Konzepte eingeführt, die die 3D-Interaktion mit 3D-Eingabegeräten unterstützen. Ergänzt werden die Visualisierungshilfen durch Konzepte zur Steigerung der visuellen Rückkopplung während laufender Interaktionen (vgl. Kapitel 4.2).

4.1.1 Tiefenhinweise und ihre Anwendbarkeit in Modellierungssystemen

In Modellierungssystemen kommen die in Kapitel 2.4 vorgestellten Form- und Tiefenhinweise kaum zum Einsatz, stattdessen wird aus Geschwindigkeitsgründen auf den Drahtgittermodus oder eine einfache, schattierte Darstellung zurückgegriffen. Nachteilig ist die dadurch erschwerte und somit verlangsamte Wahrnehmung räumlicher Strukturen. Um die räumliche Struktur erkennen zu können, muß auf Effekte wie ’Verdeckung’ und ’Bewegungsparallaxe’ zurückgegriffen werden, d.h. die virtuelle Kamera bzw. das Modell muß bewegt werden, nur um dessen räumliche Struktur erkennen zu können, was einen erhöhten und teilweise unnötigen Interaktionsaufwand darstellt.

Bei Einsatz von 3D-Eingabegeräten mit graphischem 3D-Echo, ist es höchst unerwünscht, das Modell stets drehen zu müssen, um die relative Lage von Modell und 3D-Echo erkennen zu können. Es muß also eine Möglichkeit geschaffen werden, räumliche Zusammenhänge auch ohne Bewegung aus der Szene extrahieren zu können. 3D-Ausgabe und Schattenwurf sind naheliegende und für Modellierungsanwendungen nutzbare Visualisierungstechniken, die allerdings den Berechnungs- und Darstellungsaufwand z.T. beträchtlich erhöhen. Materialinformation stellt für CAD-Modelle wichtige Semantik bereit, z.B. für FEM-Analysen. Ein Material definiert auch die Oberflächenstruktur (Textur) eines Objektes. Material und Textur kann bereits im Konstruktionsprozeß die Realitätsnähe von Objekten steigern. Hardwareunterstützung vorausgesetzt, ist sogar direkt-manipulatives Modellieren mit texturierten Objekten möglich.

Jedoch lassen sich nicht alle der in Kapitel 2.4 genannten visuellen Hinweise in Modellierungsanwendungen sinnvoll einsetzen. Im Rahmen dieser Arbeit durchgeführte Experimente mit atmosphärischen Effekten haben ergeben, daß diese von den Testpersonen nicht als Tiefenhinweise akzeptiert werden, sondern die Testpersonen eine mit atmosphärischen Effekten angereicherte Szene als ’unscharf’ beurteilen. Akkomodation ist ein anderer Effekt, der in computergenerierten Szenen nicht genutzt werden kann, da immer auf die Bildebene des darstellenden Gerätes akkomodiert wird. Der Widerspruch zwischen dem Abstand des virtuellen Objektes zum Betrachter und der Akkomodationsebene ist im Extremfall sogar ein Grund dafür, daß der 3D-Effekt bei stereoskopischer Ausgabe zusammenbricht.

4.1.2 Visualisierungstechniken und -hilfen zur Unterstützung der 3D-Modellierung

Unter Beachtung existierender Erkenntnisse aus der Wahrnehmungspsychologie und Untersuchungen im Bereich computer-generierter Szenen sowie Erfahrungen aus Benutzertests wurden folgende Visualisierungstechniken und -hilfen für benutzer-zentrierte Modellierungssysteme im

Rahmen der hier vorgestellten Arbeit ausgewählt, weiterentwickelt und prototypisch implementiert.

- Objekte werden standardmäßig schattiert dargestellt.
  Während der Objekterzeugung ist eine schattierte oder semi-transparente Darstellung möglich. Aufgrund der Vorteile, die eine Texturierung bezüglich der Form- und Tiefenwahrnehmung bietet, kann schon während der graphisch-interaktiven, direkt-manipulativen Objekterzeugung mit Texturen gearbeitet werden; aus Performanzgründen ist eine Hardwareunterstützung erforderlich. Es hat sich jedoch als vorteilhaft erwiesen, während der interaktiven Objekterzeugung mit einer Drahtgitterdarstellung zu arbeiten, da sie das Zentrum des 3D-Echos unter keinen Umständen verdeckt; nach Abschluß der interaktiven Objekterzeugung wird automatisch auf eine schattierte Darstellung gemäß dem ausgewählten Material umgeschaltet.
- Die Szene ist von einem virtuellen Konstruktionsraum umgeben.
  Die Einführung einer Hysteresisschleife zur Steuerung des Verhaltens des Konstruktionsraumes sorgt für eine dynamische, aber in Grenzen stabile Anpassung der Abmaße des Raumes an die Szene.
- Die Objekte werfen transparente Schatten auf die Wände des umgebenden virtuellen Konstruktionsraumes.
  Die Transparenz gestattet es, aus einem Schatten, der einem Röntgenbild ähnelt, Informationen über die innere Struktur des schattenwerfenden Objektes abzuleiten; auch Überlappungen von Objekten werden so auf einfache Weise sichtbar.
- Als graphisches 3D-Echo kommt ein 3D-Fadenkreuz zum Einsatz, das den gesamten virtuellen Konstruktionsraum durchspannt (*full space cursor*).
  Die Achsen des Cursors durch den Raum und Fußpunkte auf den Wänden machen seine Position bezüglich der Objekte in der Szene deutlich. Im Stereobetrieb kann die dem Benutzer zugewandte Achse automatisch gekappt werden, um Irritationen durch die hohe Disparität der dem Betrachter entgegenkommenden Achse zu vermeiden.
- ’Center Shapes’ im Zentrum des 3D-Cursors.
  Verschiedene ’Center Shapes’ im Zentrum des 3D-Cursors verdeutlichen dem Benutzer den Interaktionsstatus sowie aktivierte Constraints, Diskretisierungswerte, etc. Das Konzept der ’Center Shapes’ stellt sowohl eine Erweiterung des 2D-Mauszeigers auf 3D dar als auch eine Generalisierung hinsichtlich der Funktionalität eines graphischen Echos, das hier nicht nur den Dialogstatus anzeigt, sondern z.B. auch erlaubte Modifikationen und deren Wertebereiche (siehe Kapitel 4.2.9.3).
- Eine Pickkugel im Zentrum des 3D-Cursors.
  Die Pickkugel bildet seitens der Visualisierung die Basis für ein echtes 3D-Picking (siehe Kapitel 4.2.10). Eine transparente Darstellung der Pickkugel macht Durchdringungen von 3D-Cursor und Objekten offensichtlich.
- Die Szene kann stereoskopisch dargestellt werden.

Abbildung 29 zeigt beispielhaft einige der o.g. Visualisierungstechniken und visuellen Form- und Tiefenhinweise. In der prototypischen Implementierung in Form des Systems ARCADE können die verschiedenen Visualisierungselemente ein- bzw. ausgeschaltet und somit nahezu beliebig kombiniert werden, um eine weitgehende Anpaßbarkeit an die unterschiedlichen Anforderungen verschiedener Hardware-Umgebungen bzw. Benutzerpräferenzen zu ermöglichen. Beispielsweise können die Wände entlang jeder Hauptachse deaktiviert werden oder die Schatten können als Drahtgittermodell und auch nicht-transparent dargestellt werden. Von diesen Konfigurationsmöglichkeiten wurde auch im Rahmen der Evaluierung der Visualisierungstechniken Gebrauch

gemacht, wobei eine Teilmenge der genannten Visualisierungstechniken im Hinblick auf bestimmte Interaktionsaufgaben untersucht wurde (siehe Kapitel 5.2).

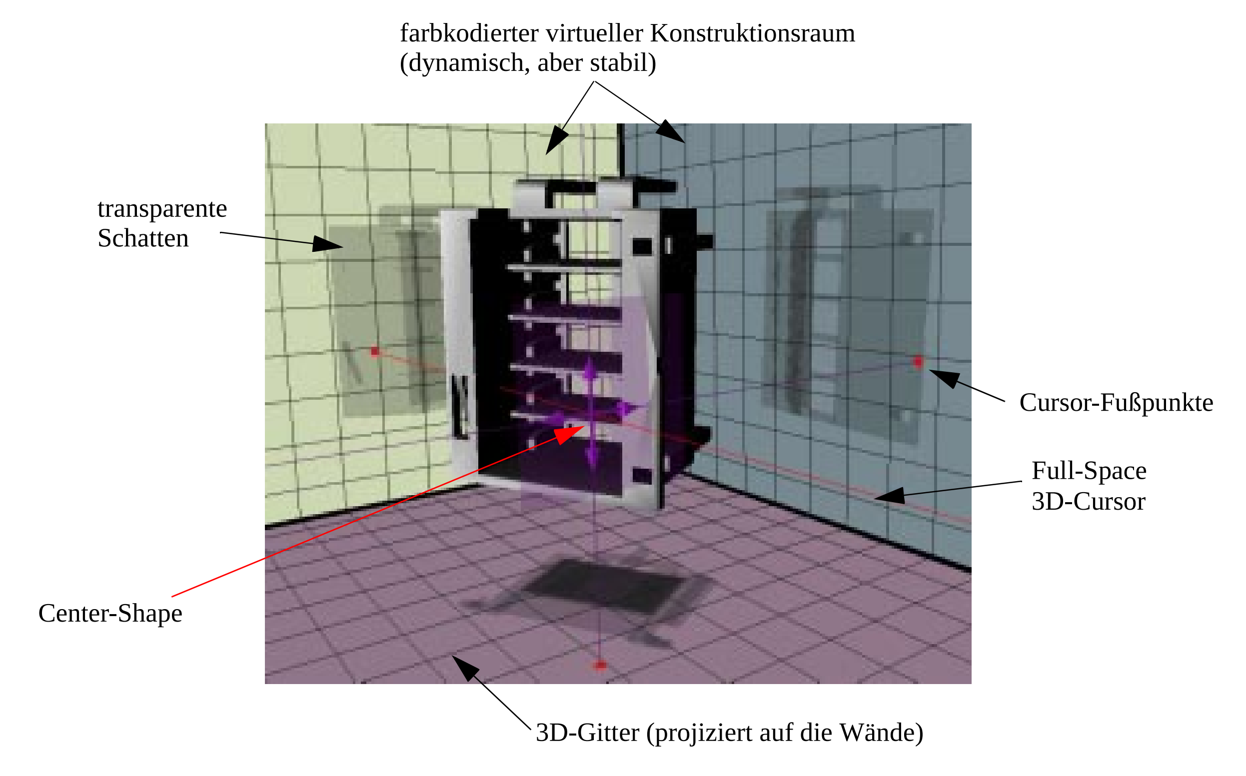


Abbildung 29: Visualisierungshilfen und -techniken

4.1.2.1 Intelligenter virtueller 3D-Konstruktionsraum

Die Idee, den Konstruktionsraum mit sichtbaren, virtuellen Wänden zu umgeben, geht auf [107] zurück. Genesis [58] war eines der ersten Modellierungsanwendungen aus der Forschung, das diese Idee aufgriff und umsetzte. Als kommerzielles Softwaresystem, das ebenfalls Wände nutzt, um den 3D-Raum darzustellen, in dem interagiert wird, ist Showcase von SGI [142] zu nennen. Allen bekannten Ansätzen ist gemein, daß die Wände entweder zu statisch sind oder zu nervös reagieren. In Genesis mußte die Größe des Konstruktionsraumes manuell vorgegeben werden. Das Erzeugen von Objekten außerhalb des Raumes war nicht möglich. Das Ändern der Größe des Raumes war nur umständlich durch alpha-numerische Werteingabe möglich. In Showcase hat der Benutzer die Wahl zwischen einem statischen Verhalten des Raums oder einem automatischen Mitwachsen bei Vergrößerung der Szene. Das Anpassen an eine kleinere Szene erfolgt hingegen nur auf Benutzeraktion.

In einem CAD-System ist die Größe der Szene während des Konstruktionsprozesses eines Bauteils sehr variabel. Hilfsgeometrie und subtraktive Operationen führen dazu, daß die Szene in ihren Ausmaßen sowohl wachsen als auch schrumpfen kann. Daher ist es sinnvoll, daß sich der umgebende Konstruktionsraum, d.h. die ihn visualisierenden Wände, dynamisch in beide Richtungen anpaßt. Die direkte Ankopplung der Wände an jede Szenenänderung führt - wie Experimente im Rahmen dieser Arbeit gezeigt haben - zu einem sehr unruhigen und für den Betrachter inakzeptablen Bildaufbau. Durch Einführung eines Hysterese-Effektes, der das Verhalten der Wände stabilisiert, kann ein in beide Richtungen entlang der drei Achsen des Weltkoordinatensystems dynamisches Verhalten realisiert werden, das nicht zur Nervosität neigt

und somit den Benutzer nicht irritiert. Der Benutzer kann die Hysteresisschleife parametrisieren und somit Einfluß auf das dynamische Verhalten des Raumes nehmen.

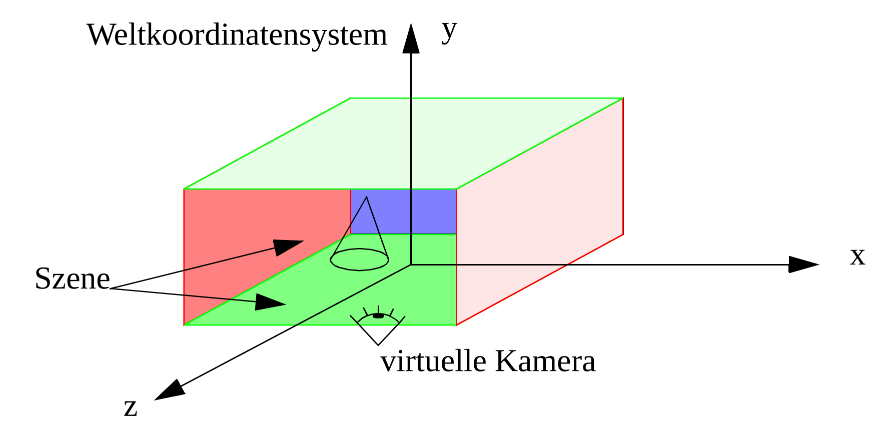


Abbildung 30: Der virtuelle Konstruktionsraum

Die Wände des virtuellen Konstruktionsraumes stehen senkrecht auf den Achsen des Weltkoordinatensystems. Die Farbkodierung der Wände erleichert dem Betrachter die Zuordnung zu den Achsen des Weltkoordinatensystems (x,y,z korrespondiert mit r,g,b) und verbessert die Orientierung von einem beliebig gewählten Blickpunkt aus (siehe Abbildung 30). Die virtuelle Kamera ist ebenso Bestandteil der Szene wie das Zentrum des 3D-Cursors. Bei jeder Änderung innerhalb der 3D-Szene oder bei jeder Kamerabewegung wird die BoundingBox um die Szene neu berechnet. Aus der Größe der BoundingBox ergibt sich die Größe des Raumes wie nachfolgend dargestellt.

Eine BoundingBox-Berechnung um die Elemente der Szene liefert die Szenegrenzen in Weltkoordinaten:

$$bb_{x_{min}} = \text{Minimum der BoundingBox in x-Richtung}$$
$$bb_{x_{max}} = \text{Maximum der BoundingBox in x-Richtung}$$
$$bb_{y_{min}} = \text{Minimum der BoundingBox in y-Richtung}$$
$$bb_{y_{max}} = \text{Maximum der BoundingBox in y-Richtung}$$
$$bb_{z_{min}} = \text{Minimum der BoundingBox in z-Richtung}$$
$$bb_{z_{max}} = \text{Maximum der BoundingBox in z-Richtung}$$

Daraus läßt sich für jede Achse das Zentrum der BoundingBox berechnen:

$$\forall\, i \in \{x, y, z\} : \qquad bb_{i_{cen}} = \frac{bb_{i_{min}} + bb_{i_{max}}}{2}$$

Durch Einführung einer Hysteresisschleife (siehe Abbildung 31) wird die Dynamik der Wände gedämpft, die Möglichkeit der automatischen Anpassung der Wände sowohl bei Vergrößerung als auch bei Verkleinerung der Szene bleibt aber erhalten.

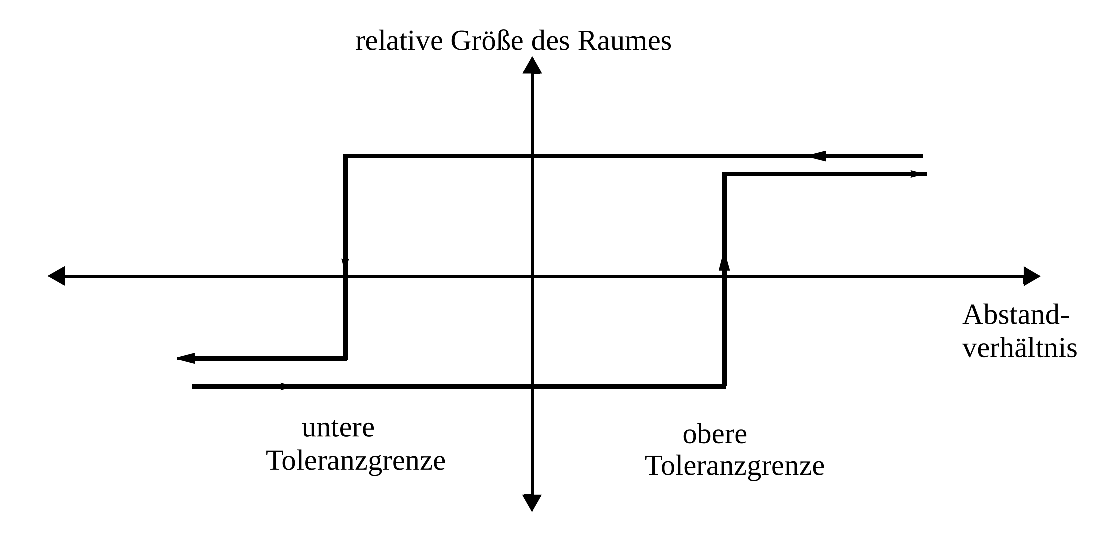


Abbildung 31: Hysterese-Effekt des virtuellen Raumes

Zwei Toleranzgrenzen t_o (obere Toleranzgrenze) und t_u (untere Toleranzgrenze) bestimmen, ob die Wandpositionen neu berechnet werden. Die obere Toleranzgrenze bestimmt die erlaubte Näherung der Szene an eine Wand des Konstruktionsraumes bis diese verschoben wird. Die untere Toleranzgrenze gibt die erlaubte, maximale Entfernung der Szene von einer Wand an; bei Überschreiten dieser Entfernung wird die Wand in Richtung des Szenenzentrums 'nachgezogen', der Raum verkleinert sich. Die Toleranzgrenzen können vom Benutzer im Bereich des offenen Intervalls] 0, 1 [gewählt werden und müssen der Bedingung $t_u < t_o$ genügen. Die Toleranzgrenzen geben an, bei welchem Verhältnis aus der Distanz der BoundingBox-Grenze zum Mittelpunkt des virtuellen Raumes zur Distanz der zugehörigen Wand zum Raummittelpunkt, die Wandposition neu berechnet wird. Dieses Verhältnis wird im folgenden auch Abstandsverhältnis *av* genannt.

Die Abstandsverhältnisse werden bei jeder Änderung der BoundingBox wie folgt berechnet:

$$\forall\, i \in \{x, y, z\} : \qquad av_{i_{min}} = \frac{r_{i_{cen}} - bb_{i_{min}}}{r_{i_{cen}} - w_{i_{min}}}$$

$$\forall\, i \in \{x, y, z\} : \qquad av_{i_{max}} = \frac{bb_{i_{max}} - r_{i_{cen}}}{w_{i_{max}} - r_{i_{cen}}}$$

mit:

$av_{i_{max}}$ = Abstandsverhältnis bei Maximalwert in Richtung i
$av_{i_{min}}$ = Abstandsverhältnis bei Minimalwert in Richtung i
$w_{i_{max}}$ = obere Wandposition in Richtung i
$w_{i_{min}}$ = untere Wandposition in Richtung i
$r_{i_{cen}}$ = Raumzentrum in Richtung i

Eine neue Position für jeweils eine der sechs Wände wird nur dann berechnet, wenn das entsprechende Abstandsverhältnis die obere Toleranzgrenze überschritten bzw. die untere Toleranzgrenze unterschritten hat. Mit Hilfe eines Benutzer-definierten Faktors, der bestimmt,

wieviel größer der Raum als die BoundingBox sein soll, ergeben sich die Wandpositionen auf jeder Achse wie folgt:

$$\forall\, i \in \{x, y, z\} : \qquad \text{wenn } av_{i_{min}} < t_u \text{ , dann } w_{i_{min}} = bb_{i_{cen}} - g_{rel} \cdot (bb_{i_{cen}} - bb_{i_{min}})$$

$$\forall\, i \in \{x, y, z\} : \qquad \text{wenn } av_{i_{max}} > t_o \text{ , dann } w_{i_{max}} = bb_{i_{cen}} + g_{rel} \cdot (bb_{i_{max}} - bb_{i_{cen}})$$

$$\text{mit } g_{rel} = \text{ relative Größe des Raums}$$

Ein abschließendes Beispiel soll dies erläutern:
Ist die obere Toleranzgrenze auf den Wert 0,8 gesetzt, so wird bei jeder Änderung der BoundingBox um die Szene zunächst das Abstandsverhältnis auf der zugehörigen Achse ermittelt. Sobald die Entfernung einer Fläche der BoundingBox zum Raumzentrum mehr als 80% des Abstands der Wand zum Raumzentrum einnimmt, wird die neue Wandposition in diese Richtung unter Beachtung der gewünschten relativen Größe des Raumes um die BoundingBox neu berechnet und entsprechend visualisiert (vgl. Abbildung 32); äquivalent wird bei Unterschreiten der unteren Toleranzgrenze verfahren. Anzumerken bleibt, daß nicht nur die Wandposition, sondern auch die Ausmaße der benachbarten Wände neu berechnet werden müssen, um eine korrekte Visualisierung zu erhalten [99].

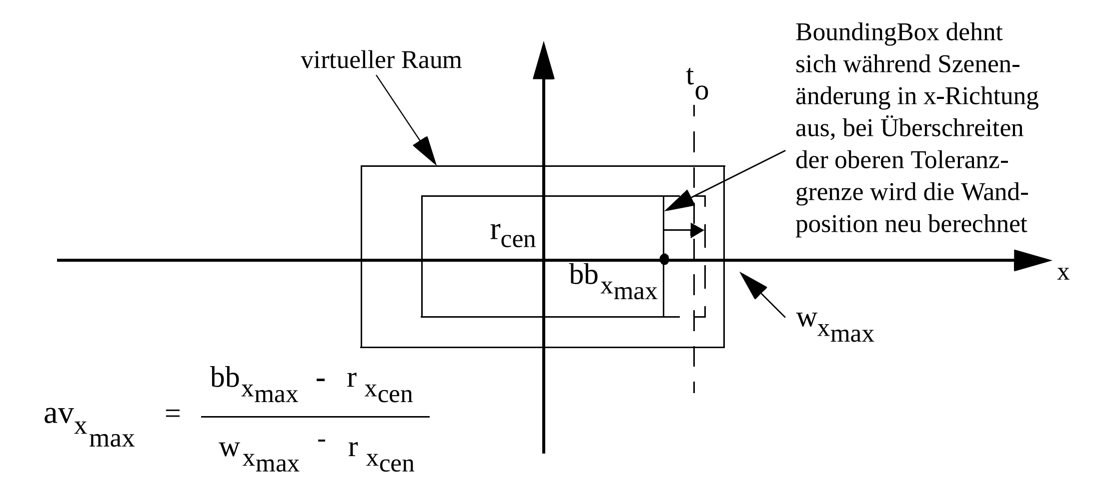


Abbildung 32: Neuberechnung einer Wandposition des virtuellen Konstruktionsraumes

Mit dem dargestellten Mechanismus kann der Benutzer durch Wahl der Parameter t_u und t_o sowie der relativen Größe des Raumes g_{rel} auf einfache Weise steuern, wie sensibel der virtuelle Raum auf Änderungen innerhalb der Szene reagieren soll. Beispielsweise reagiert der Raum bei einer hohen oberen und einer niedrigen unteren Toleranzgrenze recht träge auf Szenenänderungen. Obwohl sich das Verhalten der Wände mit dem vorgestellten Verfahren sehr feinfühlig steuern läßt, ist es manchmal wünschenswert, eine Wand interaktiv auf eine bestimmte Position zu setzen und dort bis auf weiteres einzufrieren. Dies ist z.B. dann der Fall, wenn die Schatten eines Objektes auf einer Wand nicht mehr sichtbar sind, da diese sich zu weit vom Objekt entfernt hat. Ein solches Verhalten ließe sich in Ergänzung zu dem vorgestellten Mechanismus problemlos implementieren.

4.1.2.2 3D-Gitter

Gitternetzpunkte oder Gitternetzlinien dienen im allgemeinen dazu, den Mauszeiger in Richtung der Hauptachsen in äquidistanten Schritten zu bewegen. In 2D-Systemen oder auch in 2-dimensionalen Ansichten in 3D-Systemen können Gitter als Linien oder deren Schnittpunkte als Punkte dargestellt werden. Übertragen auf 3D, ist eine Visualisierung des Gitters direkt im 3-dimensionalen Raum zwar naheliegend, führt aber zu einer Überfrachtung der Szene, so daß der Blick auf das Wesentliche nicht mehr gewährleistet ist.

Die Visualisierung des Konstruktionsraumes ermöglicht es, ein 3D-Gitter zu realisieren und zu visualisieren, ohne die Szene zu überfrachten. Zu diesem Zweck werden die Gitternetzlinien orthogonal auf die Wände projiziert (siehe Abbildung 33). Bei diesem Vorgang ist zu beachten, daß - unabhängig von der Raumposition einer Wand - die Gitternetzlinien stets relativ zum Ursprung des Weltkoordinatensystems auszurichten sind. Bei der Anpassung der Position einer Wand sowie der Abmessungen ihrer Nachbarwände müssen die Gitternetzlinien auf den Wänden sowohl in ihrer Lage als auch in ihrer Länge neu berechnet werden, um eine korrekte Visualisierung zu erreichen [99]. Zu beachten ist, daß die Gitternetzlinien nicht Bestandteil der Szene sind, um die die Bounding Box zur Bestimmung der Wandpositionen des virtuellen Konstruktionsraumes berechnet wird; die Gitternetzlinien können nicht gleichzeitig auf einer virtuellen Wand liegen und Bestandteil der Szene sein, wenn die Wand einen definierten Abstand zur Szene hat.

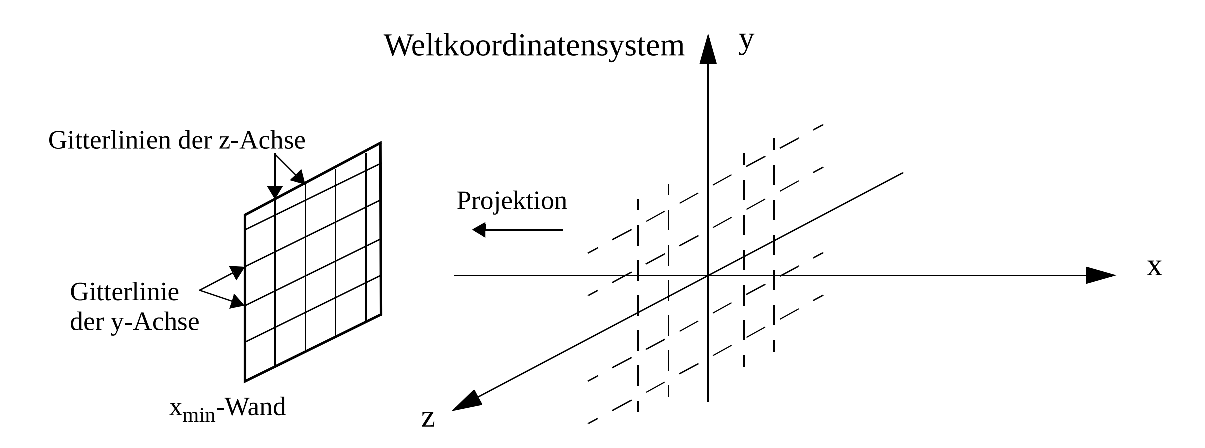


Abbildung 33: 3D-Gitterprojektion auf die x-Wand [99]

Das 3D-Gitter kann eingesetzt werden, um die Bewegungen des 3D-Cursors im Raum zu diskretisieren. In der Implementierung wurde die Möglichkeit realisiert, die Rasterbreiten in jeder Richtung individuell einzustellen. Weiterhin ist wählbar, ob das Gitter für eine bestimmte Richtung/Achse dargestellt werden soll oder nicht und ob der Cursor sich nur diskret von Gitterschnittpunkt zu Gitterschnittpunkt oder auch kontinuierlich im Raum bewegen darf. Durch den Full-Space-Cursor inkl. seiner Fußpunkte läßt sich problemlos wahrnehmen, auf welchem Gitterkreuzungspunkt der 3D-Cursor steht. Neben einer Kontrolle des Cursors eignet sich das Gitter - bei bekannter Rasterbreite - in Verbindung mit Schattenwurf auch dazu, die Größe von Objekten abzuschätzen.

4.1.2.3 Schattenwurf

Aus der Wahrnehmungspsychologie (siehe Kapitel 2.4) ist bekannt, daß Schatten ausgezeichnete Hinweise auf die räumlichen Beziehungen zwischen Objekten bieten. Die Berechnung von Schatten in computer-generierten Szenen stellt indes hohe Anforderungen an die Leistungsfä-

higkeit des Rechners. Dies gilt umso mehr, je komplexer und interaktiver die Szenen sind, da mit jeder Bewegung eines Objektes geprüft werden muß, welche Schatten aktualisiert werden müssen. Da Modellierungssysteme durch Interaktivität und Komplexität der Modelle gekennzeichnet sind, stellt die Integration von Schattenwurf in ein Modellierungssystem eine Herausforderung insbesondere an die Hardware dar.

Aus der Computergraphik sind verschiedene Verfahren zur Erzeugung von Schatten in einer Szene bekannt:

- Projektion eines Objektes auf eine Ebene [22],
- Schattenvolumen, z.B. mit Hilfe von Binary Space Partitioning Trees [35],
- Multipass-Rendering [119], [136] und
- Radiosity [102] sowie Echtzeit-Radiosity [132].

Alle genannten Verfahren benötigen Leistungsmerkmale, die bei hinreichend komplexen Modellen von heutigen CAD-Graphikworkstations nur bedingt erfüllt werden. Aus diesem Grund wurde ein Ansatz gewählt, der einen Quasi-Schattenwurf auf die Wände des Konstruktionsraumes darstellt und das Konzept der multiplen Ansichten in einer Ansicht integriert. Zwischen den Objekten in der Szene findet kein Schattenwurf statt - zu evaluieren wäre, ob ein Schattenwurf zwischen Objekten und die damit verbundenen Abschattungen bestimmter Bereiche in der Szene dem Modellierungsprozeß überhaupt dienlich wären oder ob sie zu Irritationen bei dem Betrachter führen.

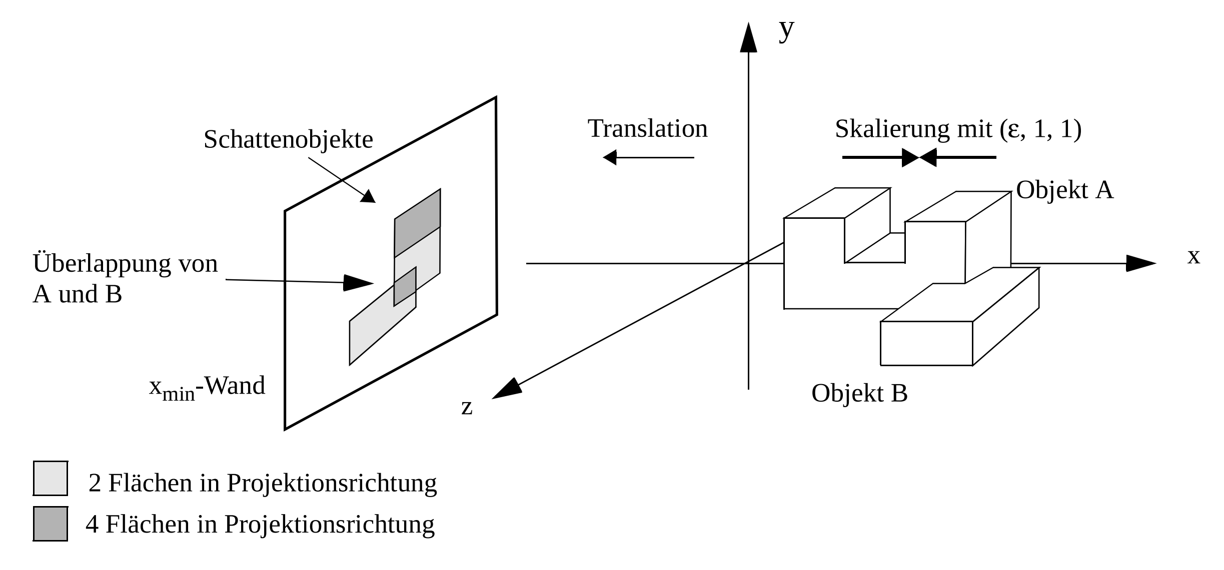


Abbildung 34: Schema der Schattenerzeugung

Ein Schatten eines Objektes wird dadurch erzeugt, daß das Objekt in der Projektionsrichtung durch Skalierung mit einem sehr kleinen Wert verschieden von Null kollabiert wird und dann in die Ebene der Wand transliert wird, um dort erneut dargestellt zu werden (siehe Abbildung 34), d.h. jeder Schatten erfordert das nochmalige Darstellen des gesamten Objektes, was insbesondere zu Lasten der Graphikhardware geht - durch Verwendung von *levels-of-detail* kann die Last verringert werden. Um Flickerprobleme durch die beschränkte Auflösung des z-Buffers zu vermeiden, wird der Schatten geringfügig vor der Wand dargestellt. Die Position der Schattenobjekte hängt sowohl von der Position der Objekte als auch der Wände ab. Ändert sich eine Objekt- oder Wandposition, so werden die Schattenobjekte nachgezogen. Wie die Gitternetzlinien, dürfen auch die Schattenobjekte nicht Bestandteil der Szene sein, um die die BoundingBox berechnet wird, von der die Wandpositionen des Konstruktionsraumes abhängen.

Durch transparentes Darstellen der Schatten entsteht der Eindruck eines Röntgenbildes von dem schattenwerfenden Objekt. Aus den Graustufen eines transparenten Schattens lassen sich Rückschlüsse über die innere Struktur des Objektes ziehen. In den Regionen, wo sich in der Projektionsrichtung mehr Flächen überlappen, entsteht ein dunklerer Farbton. Allerdings entscheidet nicht die Materialdicke oder -beschaffenheit über das Erscheinungsbild im Schatten, sondern 'nur' die Anzahl der sich überlappenden und überlagernden Flächen. Auch Überlappungen verschiedener Objekte in Projektionsrichtung lassen sich so leicht erkennen (siehe Abbildung 34).

4.1.2.4 Der 3D-Full-Space-Cursor

Als graphisches Echo sind Mauszeiger, Fadenkreuze oder virtuelle Hände in VR-Systemen stark verbreitet. Fadenkreuze überspannen herkömmlicherweise ein Bildschirmfenster. Der 3D-Full-Space-Cursor stellt ein Fadenkreuz in 3D dar und geht auf [107] zurück. Er durchspannt den gesamten Konstruktionsraum.

Im Rahmen dieser Arbeit durchgeführte Experimente zeigen, daß es bei der Darstellung eines 3D-Full-Space-Cursors zu Zweideutigkeiten kommt: Aus manchen Blickwinkeln kann der Betrachter bei monoskopischer Ausgabe nicht mit Bestimmtheit feststellen, welches Ende einer Achse ihm zugewandt ist. Durch Visualisierung der Schnittpunkte zwischen Full-Space-Cursor und Wänden läßt sich dieses Problem lösen. Bereits in der Arbeit von Nielson und Olson wurde die Idee erwähnt, die Schnittpunkte - die sog. Fußpunkte - zu visualisieren, allerdings implementierten sie Fußpunkte nicht, um ihre Hypothese vom Nutzen der Fußpunkte nachzuweisen.

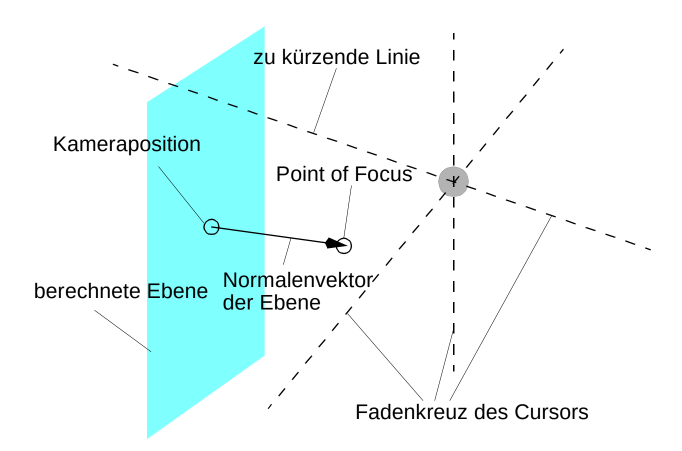


Abbildung 35: Kürzung des Full-Space-Cursors im Stereomodus

Ein weiteres Problem beim Einsatz eines Full-Space-Cursors tritt bei stereoskopischer Ausgabe auf: Da sich der Cursor über den gesamten Raum erstreckt, wird die dem Betrachter zugewandte Achse des Cursors stark disparat dargestellt. Das visuelle System kann die divergenten Bilder nicht mehr zu einer 3D-Gestalt kombinieren, wodurch der 3D-Eindruck der gesamten Szene stark beeinträchtigt wird, wenn nicht sogar zusammenbricht. Dieser Effekt ist umso gravierender, je großflächiger das Ausgabegerät ist. Abhilfe schafft ein Mechanismus, der die dem Benutzer zugewandte Achse des Cursors kappt. Um dies zu realisieren, muß festgestellt werden, welches der sechs Linienenden hinter der Kameraebene liegt und gekürzt werden muß. Mit Hilfe der Kameraposition und der Blickrichtung wird diese Ebene berechnet. Nun kann entschieden werden, welcher Linienendpunkt sich hinter der Kamera befindet (siehe Abbildung 35). Das Linienstück bis zum Zentrum des Cursors kann nun komplett oder partiell eliminiert werden. Die

Kürzung einer Cursorachse beeinträchtigt die positiven Eigenschaften des Full-Space-Cursors kaum.

Steuerung des 3D-Full-Space-Cursors über 3D-Eingabegeräte

Der Full-Space-Cursor wird durch Anwenden, der von einem 3D-Eingabegerät stammenden Ereignisse, im Raum bewegt. Die Möglichkeit, den 3D-Cursor durch Anwenden von Rotationsereignissen um sein Zentrum zu drehen, hat sich nicht als adäquat erwiesen, da die Korrespondenz zwischen Cursorzentrum und Fußpunkt sowie Cursorkoordinatensystem und Weltkoordinatensystem verlorengeht. Dies übersteigt sowohl die kognitiven als auch die motorischen Fähigkeiten der meisten Benutzer. Anzumerken ist, daß die anliegende Rotation für Objektmanipulationen genutzt wird und mit Hilfe der Center-Shapes (siehe Kapitel 4.1.2.5) am 3D-Cursor auch ohne laufende Objektmanipulation visualisiert werden kann.

Unterschiedliche 3D-Eingabegeräte erfordern eine unterschiedliche Interpretation der gelieferten Werte. Bei Verwendung eines relativ arbeitenden 3D-Eingabegerät, das zu einem Zeitpunkt einen Translationsvektor t = (x,y,z) liefert, ergibt sich die Cursorposition c_i zu einem Zeitpunkt i zu:

$$c_i = \sum_{j=0}^{i} t_j$$

Wird ein absolut arbeitendes Eingabegerät verwendet, so ist die Cursorposition durch die Position des 3D-Eingabegerätes im Raum definiert - i.d.R. ist eine Transformation zwischen dem Koordinatensystem des absoluten 3D-Eingabegerätes und dem Koordinatensystem, in dem sich der 3D-Cursor befindet, erforderlich.

Auch die von einem relativ arbeitenden 3D-Eingabegerät gelieferten Werte beziehen sich auf das ihm eigene Koordinatensystem. Der Benutzer eines Modellierungssystems sollte die Möglichkeit zur freien Wahl der Betrachtungsposition haben. Würde man die Translationswerte nach obiger Formel direkt auf den Cursor anwenden, so wäre die Reiz-Reaktions-Korrespondenz aus nahezu allen Betrachtungswinkeln nicht mehr gegeben.

Ein Beispiel soll dies verdeutlichen: In der in Abbildung 36 dargestellten Situation bewegt der Benutzer die Kappe der Space-Mouse nach rechts in x-Richtung ihres Koordinatensystems. Werden nun die relativen Translationen direkt auf den 3D-Cursor übertragen und die Szene von Kameraposition A aus betrachtet, bewegt sich der 3D-Cursor wie erwartet nach rechts in Richtung der x-Achse. Betrachtet der Benutzer die Szene nun von Kameraposition B aus und führt dieselbe Bewegung nach rechts aus, so bewegt sich der Cursor ebenfalls in Richtung der positiven x-Achse, aber diesmal vom Betrachtungsstandpunkt nach links! Die Reiz-Reaktions-Korrespondenz ist nicht mehr gegeben.

Die Reiz-Reaktions-Korrespondenz kann wiederhergestellt werden, wenn jeder Translationsvektor t_i zunächst mit der zum Zeitpunkt i gültigen Kamerarotation R_i multipliziert wird. Daraus ergibt sich folgende, die Reiz-Reaktions-Korrespondenz wahrende, Formel zur Berechnung der Cursorposition c zum Zeitpunkt i:

$$c_i = \sum_{j=0}^{i} R_j \cdot t_j$$

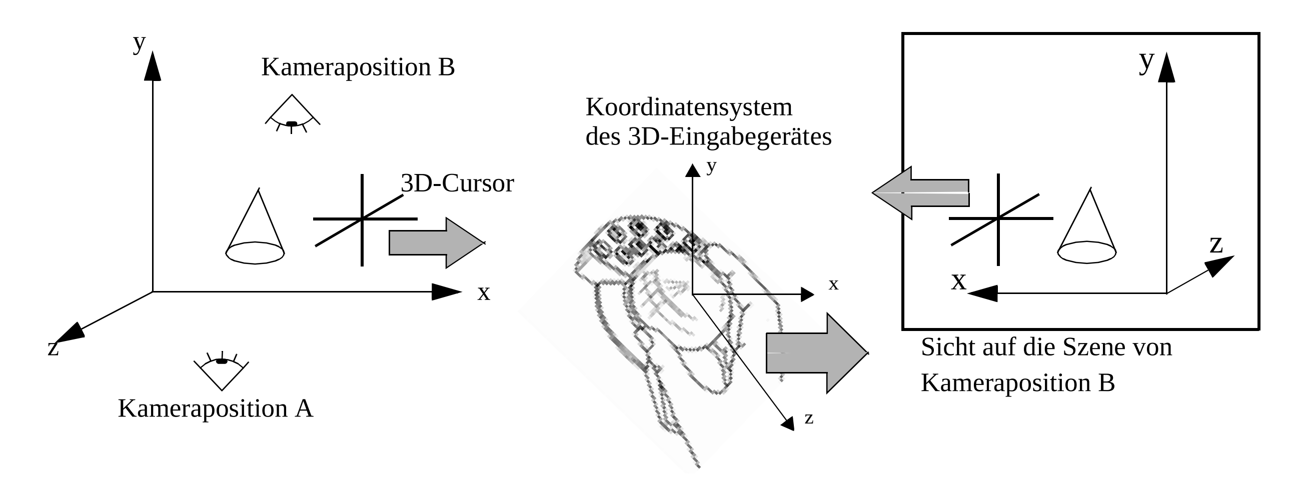
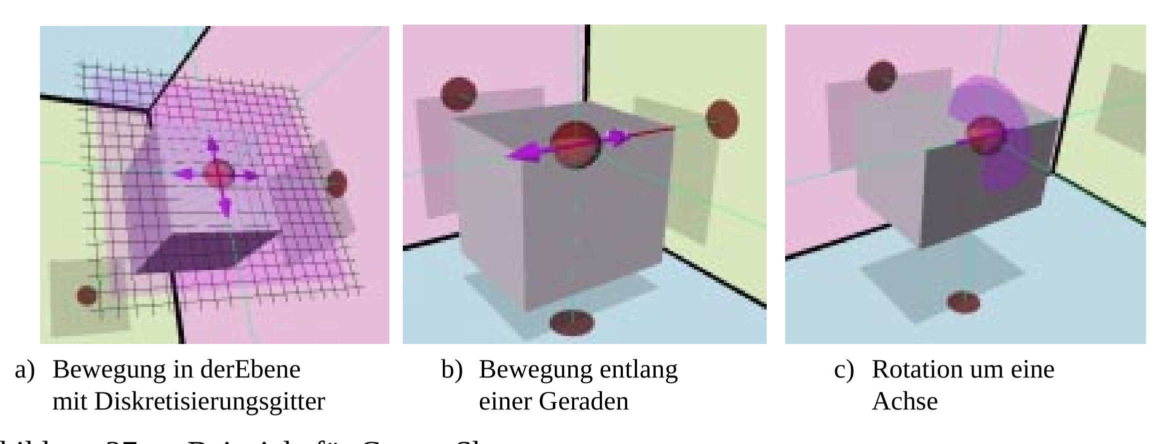


Abbildung 36: Abbildung der Ereignisse des 3D-Eingabegerätes in den Kameraraum

4.1.2.5 Center-Shapes

Center Shapes sind unterschiedliche Geometrien, die kontext-abhängig im Zentrum des 3D-Cursors dargestellt werden (siehe Abbildung 37). Sie können als Äquivalent zu den aus 2D-Systemen bekannten unterschiedlichen Mauszeigern betrachtet werden, erweitern die dortigen Möglichkeiten aber beträchtlich. Center Shapes können sich mit dem 3D-Cursor bewegen oder aber bei ihrer Aktivierung an der initialen Position verharren, so daß sich der 3D-Cursor entlang bzw. innerhalb deren Geometrie bewegen kann - Center Shapes weisen also ein bestimmtes Verhalten auf. Center Shapes werden i.d.R. semi-transparent dargestellt, um die Sicht auf die Szene nicht zu stark zu beeinträchtigen.



a) Bewegung in derEbene mit Diskretisierungsgitter
b) Bewegung entlang einer Geraden
c) Rotation um eine Achse

Abbildung 37: Beispiele für Center-Shapes

Folgende kontext-sensitive Center Shapes mit unterschiedlichem Verhalten wurden realisiert:

- Visualisierung der Cursor-Constraints: Bewegen in einer Ebene und entlang einer Linie (siehe Abbildung 37),
- Visualisierung der erkannten Geste bei der Objektmodifikation (siehe Kapitel 4.2.6),
- Visualisierung der Diskretisierung bei der Objektmanipulation (siehe Kapitel 4.2.9),
- Visualisierung der Parametrisierung bei der Objektmanipulation (siehe Kapitel 4.2.9.3) und
- Visualisierung des Pickradius in 3D in Form einer transparenten Kugel (siehe Kapitel 4.2.10.1).

Die Center Shapes sind stark mit der entsprechenden Interaktionstechnik verwoben, so daß auf sie an gegebener Stelle nochmals eingegangen wird.

Die bislang vorgestellten Visualisierungstechniken und -hilfen steigern zum einen die Form- und Tiefenwahrnehmung in der Szene, zum anderen ermöglichen erst sie echte 3D-Interaktion mit 3D-Eingabegeräten in einem Konstruktionsraum. Die im Rahmen dieser Arbeit entwickelten 3D-Interaktionstechniken und deren Visualisierungshilfen werden im folgenden Unterkapitel detailliert beschrieben.

4.2 3D-Interaktionstechniken

In diesem Unterkapitel werden die im Rahmen dieser Arbeit entwickelten 3D-Interaktionstechniken zur schnellen, präzisen Modellerzeugung und -modifikation sowie zur interaktiven Zusammenbausimulation mit 3D-Eingabegeräten dargestellt.

Mit konventionellen 2D-Eingabegeräten müssen 3D-Interaktionen, wie das Positionieren und Dimensionieren von Objekten, in eine Sequenz von 2D-Interaktionen zerlegt und vom Benutzer schrittweise durchgeführt werden. Dies erfordert eine mentale Abbildung und Zerlegung der Prozesse auf die Eingabemöglichkeiten des verwendeten Systems. Im Gegensatz dazu ermöglichen 3D-Eingabegeräte die direkte und intuitive Ausübung echter 3D-Interaktionen.

Für den Einsatz von 3D-Eingabegeräten im Konstruktionsprozeß ist es entscheidend, die zur Verfügung gestellten Freiheitsgrade geeignet zu nutzen und somit effizientes Arbeiten zu ermöglichen. Präzises Arbeiten innerhalb graphischer Interaktionen muß durch geeignete Interaktionstechniken unterstützt werden. Bei der Entwicklung entsprechender Interaktionstechniken wurde besonderer Wert auf Intuitivität und die damit verbundene Benutzerakzeptanz gelegt.

4.2.1 Anforderungen an die Interaktion in Modellierungsanwendungen

An die Interaktion im allgemeinen und die in CAD im speziellen werden eine Reihe von Anforderungen gestellt, die in verschiedenen Veröffentlichungen und Richtlinien gesammelt wurden [2], [47], [68], [168]. Die wichtigsten Anforderungen - ergänzt um Aussagen von Benutzern in Benutzertests - sind im folgenden genannt. Interaktionen sollen:

- direkt-manipulativ,
- semantisch korrekt,
- flexibel, d.h. in beliebiger Reihenfolge durchführbar,
- intuitiv und natürlich,
- effizient, d.h. mit möglichst wenigen Benutzeraktionen durchführbar,
- in ihren Freiheitsgraden einschränkbar,
- kontext-sensitiv und
- präzise

sein. Weiterhin sollen sie

- die Objekterzeugung und (lokale) Objektmanipulation unterstützen,
- den Zugriff auf topologische Elemente erlauben und
- in der Lage sein, Modellierungsoperationen automatisch auszulösen (z.B. implizite Boolsche Operationen).

3D-Eingabegeräte mit 6 Freiheitsgraden sind dazu prädestiniert, diese Anforderungen zu erfüllen. Sie erlauben es, einen Punkt im Raum mit einer einzigen Interaktion zu bestimmen oder ein Objekt gleichzeitig zu translieren und zu rotieren, so als ob man es in der Hand hielte. Daß eine hohe Zahl von Freiheitsgraden im Konstruktionsprozeß nicht immer ein Vorteil ist, war Motivation, die Topologie-basierte Modifikationstechnik (siehe Kapitel 4.2.6) zu entwickeln.

In den bisherigen Modellierungsansätzen mit 3D-Eingabegeräten wurden deren Möglichkeiten nicht in vollem Umfang ausgeschöpft und die Spezifika von CAD zu wenig beachtet (Kapitel 2.2.2). Nachfolgend werden Interaktionstechniken, Verfahren und Algorithmen dargestellt, die die bislang existierende Lücke zwischen dem schnellen, konzeptionellen Arbeiten mit 3D-Eingabegeräten und dem konventionell durchgeführten detaillierten Design schließen.

4.2.2 Objekterzeugung im Raum

Ein 3D-Eingabegerät und ein entsprechendes 3D-Echo vorausgesetzt, kann durch Translieren des 3D-Cursors jeder beliebige Punkt im Raum 'angefahren' und durch Klicken selektiert werden. Aus dieser Tatsache ergeben sich Möglichkeiten, 2D- und insbesondere 3D-Primitive direkt im Raum - unabhängig von bereits existierenden Objekten oder definierten Arbeitsebenen - zu erzeugen.

Für die Objekterzeugung wird die Preselektion des Objekttyps gegenüber einem gesten-basierten Ansatz favorisiert, da nur so sichergestellt werden kann, daß der Objekttyp und damit die für die Interaktion wichtige Semantik vor der Direkt-Manipulation bekannt ist. Im Hinblick auf eine präzise Interaktion durch objekt-spezifische Diskretisierung der Eingangsdaten ist die Kenntnis des zu erzeugenden Objekttyps unumgänglich. Gesten-basieren Ansätzen zur Geometrieerzeugung [184] mangelt es an der Möglichkeit, Diskretisierungen der Parameter von Objekten schon während der Erzeugung vornehmen zu können, da der Objekttyp erst nach erfolgten Gesteninterpretation feststeht. Gesten-basierte Ansätze sind zudem mit einem vergleichsweise höheren Interaktionsaufwand verbunden, da Objekte nicht aus einer 3D-Interaktion abgeleitet werden können, sondern mittels mehrerer Linien skizziert werden müssen.

Die Objekterzeugung im Raum folgt dem Vorbild der BoundingBox-Methode [94]. Die hierzu entwickelten semantisch-korrekten Interaktionstechnken weisen allerdings deren Nachteile (siehe Kapitel 2.2.2) nicht mehr auf. Die Erzeugung eines Primitivs läuft prinzipiell wie folgt ab:

1. Selektion des zu erzeugenden Objekttyps
2. Positionierung des 3D-Cursors an die Stelle, wo die Erzeugung begonnen werden soll
3. Selektion des ersten Konstruktionspunktes
4. Gegebenenfalls automatische, objekttyp-abhängige Einschränkung der Freiheitsgrade des Cursors
5. Bewegen des 3D-Cursors im Raum, wobei während der Bewegung die graphische Repräsentation des Objektes in Echtzeit der Cursorbewegung angepaßt wird; zur Erzeugung exakt dimensionierter Primitive kann die Interaktion diskretisiert werden (siehe Kapitel 4.2.9.2).
6. Optionales Selektieren weiterer Punkte, z.B. beim Erzeugen eines Linienzugs
7. Beenden der graphischen Interaktion

Nach Abschluß der direkt-manipulativen Interaktion wird aus den Parametern der graphischen Objektrepräsentation eine mathematisch exakte abgeleitet und im Modeller instantiiert (vgl. Kapitel 3.3).

Volumenprimitive können mit nur 2 Klicks und einer Bewegung des 3D-Cursors im Raum positioniert und dimensioniert werden. Durch sukzessives Bewegen des 3D-Cursors und wiederholtes Klicken ist es möglich, auf einfache und schnelle Weise 2D-Primitive, wie z.B. Linien, Linienzüge und Flächen, im Raum zu erzeugen.

Im folgenden werden die Besonderheiten der semantisch-korrekten Interaktionstechniken für die Erzeugung von Volumenprimitiven beschrieben.

Im Unterschied zur BoundingBox-Methode von Liang und Green [94] werden hier die Parameter der Volumenprimitive aus einem Gummibandvektor g abgeleitet, der zwischen dem ersten Konstruktionspunkt und der aktuellen Cursorposition definiert ist. Der erste Konstruktionspunkt stellt einen typabhängigen Fixpunkt der Oberfläche eines Objektes dar. Bei einem Zylinder ist dies beispielsweise der Mittelpunkt der Grundfläche, womit verhindert wird, daß dieser Punkt während der Objekterzeugung wandert, wie es bei dem Ansatz von Liang und Green der Fall ist. Dies erleichtert auch die Objekterzeugung auf existierenden Objekten (siehe Kapitel 4.2.3).

Die semantisch-korrekten Interaktionstechniken folgen einem strengen Objektbegriff, so existiert für einen Quader eine andere Erzeugungsmethode als für einen Würfel. Bei einem Würfel wird die mögliche Cursorbewegung auf die Raumdiagonalen beschränkt, um sicherzustellen, daß die Seitenlängen gleich sind; bei einem Quader ist dies nicht der Fall [99]. Aus einem Würfel kann auch während einer anschließenden Modifikation kein Quader entstehen; die Modifikation arbeitet ebenfalls semantisch-korrekt. Diese strenge Kontextsensitivität unterstellt die Verbindlichkeit der Intention des Benutzers bei der Wahl des Objekttyps vor der Objekterzeugung. Die Intention kann durch bloße Objektmodifikation nicht geändert werden - ein explizites Ändern des Objekttyps durch den Benutzer wäre hingegen möglich.

Tabelle 3 zeigt die objektspezifischen Regeln für die Bestimmung der Parameter verschiedener Volumenprimitive aus dem Gummibandvektor g. Die Spalte mit der Überschrift 'Erster Punkt' stellt die Bedeutung des ersten Konstruktionspunktes für das entsprechende Primitiv dar.

Tabelle 3: Erzeugungsregeln für verschiedene Volumenprimitive mittels 3D-Eingabe

Objekt	Parameter*	Parameterberechnung während der Erzeugung	Erster Punkt
Quader	Höhe Breite Tiefe	die Komponenten des Gummiband-vektors (Distanz in x-, y- und z-Richtung vom ersten Konstruktionspunkt bestimmen direkt die Parameterwerte	Ecke
elliptischer Zylinder elliptisches, zyl. Prisma	$Radius_x$ $Radius_y$ Höhe		Zentrum der Grundfläche
elliptischer Konus elliptischer, prismatischer Konus			
elliptischer Kegelstumpf elliptischer, prismatischer Kegelstumpf	unterer $Radius_x$ unterer $Radius_y$ Höhe oberer $Radius_x$	oberer $Radius_x$ = unterer $Radius_x$ / 2 $\frac{\text{unterer Radius}_x}{\text{unterer Radius}_y} = \frac{\text{oberer Radius}_x}{\text{oberer Radius}_y}$	
Zylinder zylindrisches Prisma	Radius Höhe	Radius = $\sqrt{g_x^2 + g_y^2}$ Höhe = g_z mit g = Gummibandvektor	
Konus prismatischer Konus			
Kegelstumpf prismatischer Kegelstumpf	unterer Radius oberer Radius Höhe	oberer Radius = unterer Radius / 2	
Kugel	Radius	Radius = Distanz zum ersten Punkt / 2 Zentrum = Zentrum zwischen erstem Punkt und Cursor	Punkt auf der Oberfläche

*) alle prismatischen Objekttypen besitzen 'Seitenanzahl' als zusätzlichen Parameter, der standardmäßig auf 6 gesetzt ist

Um die Reiz-Reaktions-Korrespondenz während der Objekterzeugung zu wahren, ist es nicht immer sinnvoll, die drei Komponenten des Gummibandvektor 1:1 auf die Parameter des in der Erzeugung befindlichen Primitivs abzubilden; hat das Primitiv mehr oder weniger als drei Parameter ist offensichtlich eine andere Abbildungsvorschrift nötig (siehe Tabelle 3).

Aber auch bei Primitiven mit genau drei Parametern kann eine 1:1-Abbildung der Komponenten des Gummibandvektors auf die Parameter zur Beeinträchtigung der Intuitivität führen. Der zylindrische Kegelstumpf (vgl. Tabelle 3) ist ein Beispiel für ein solches Primitiv mit drei Parametern, bei dem das Prinzip der Reiz-Reaktions-Korrespondenz für eine gleichzeitige Manipulation aller drei Parameter nicht aufrecht erhalten werden kann. Für solche Primitive manipuliert die entsprechende Interaktionstechnik nur zwei Parameter, nämlich Höhe und Radius der Grundfläche. Der dritte Freiheitsgrad des 3D-Eingabegerätes müßte den Radius der Deckelfläche kontrollieren, so daß Bewegungen des 3D-Cursors in x- bzw. y-Richtung den Radius der Grundfläche bzw. den Radius der Deckelfläche ändern, was wenig intuitiv und erwartungs-konform ist.

Hat ein Primitiv mehr als drei Parameter, so werden einige der Parameter in ein festes Verhältnis zu den manipulierbaren Parametern gesetzt. Beispielsweise werden von den Parametern eines elliptischen Kegelstumpfes durch den Gummibandvektor nur die Höhe und die beiden Radien der Grundfläche bestimmt, die Radien der Deckelfläche werden als die Hälfte des entsprechenden Grundflächenradius' angenommen (vgl. Tabelle 3). Das Ausnutzen der Rotationsfreiheitsgrade des 3D-Eingabegerätes für diese Parameterdefinition ist zwar prinzipiell möglich, widerspricht aber ebenfalls dem Prinzip der Reiz-Reaktions-Korrespondenz.

Ein Zylinder ist ein Primitiv mit weniger als drei Parametern. Für diesen Fall werden die Komponenten des Gummibandvektors so interpretiert, daß die Objektsemantik gewahrt bleibt. Die Höhe ergibt sich direkt aus der z-Komponente des Gummibandvektors, der Radius aus der Distanz der x- und y-Komponente zur z-Achse (vgl. Abbildung 38).

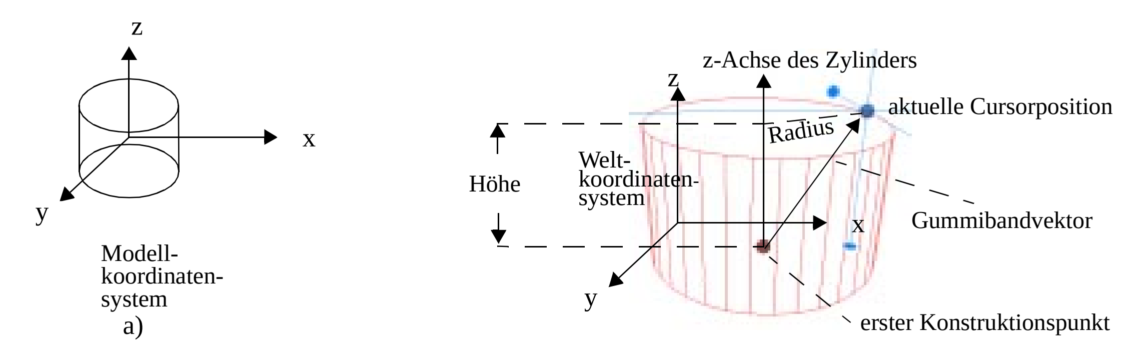


Abbildung 38: Erzeugung eines Zylinders mit einem 3D-Eingabegerät

Die Berechnung soll beispielhaft für einen Zylinder dargestellt werden. Ein Zylinder besitzt die Parameter Höhe *h* und Radius *r*. Er ist standardmäßig im Ursprung des Modellkoordinatensystems zentriert (siehe Abbildung 38a). Die Parameter *h* und *r* sowie die Translation *T* des Zylinders in das Weltkoordinatensystem ergeben sich zu einem Zeitpunkt j wie folgt.

c_0 sei die Cursorposition am ersten Konstruktionspunkt,

c_j ist die aktuelle Cursorposition und

der Gummibandvektor $g = (g_x, g_y, g_z)$ zum Zeitpunkt j ist: $g_j = c_j - c_0$.

Zur Vereinfachung wird im folgenden auf den Index j verzichtet, so daß folgt:

$$h = g_z$$

$$r = \sqrt{g_x^2 + g_y^2}$$

$$T = c_0 + \frac{1}{2} \cdot g' \qquad \text{, mit } g' = (0, 0, g_z)$$

Obwohl 3D-Eingabegeräte i.d.R. sechs Freiheitsgrade zur Verfügung stellen, wird eine Rotation während der Objekterzeugung unterbunden. Das Modellkoordinatensystem ist also stets entsprechend des Weltkoordinatensystems ausgerichtet. Benutzertests haben ergeben, daß das gleichzeitige Erzeugen, Parametrisieren und Rotieren von Primtiven kaum kontrollierbar ist und einer effizienten Interaktion eher im Wege steht als sie zu unterstützen. Dies ist zumindest teilweise auf

die motorischen Fähigkeiten des Menschen zurückzuführen, der seine Hand kaum um einen Fixpunkt rotieren kann, ohne auch eine translative Bewegung auszuführen; die Unzulänglichkeiten heutiger 3D-Eingabegeräte verschärfen dieses Problem zusätzlich. Da bei der Erzeugung auf existierenden Objekten eine automatische Orientierung gemäß der Flächennormale vorgenommen wird (vgl. Kapitel 4.2.3), ist das Fehlen der Rotationsmöglichkeit zum Erzeugungszeitpunkt eines Primitivs in der Praxis jedoch kaum relevant. Außerdem können Modifikationen, die zum Erzeugungszeitpunkt nicht angeboten oder explizit unterdrückt werden, vom Benutzer später mittels der topologie-basierten eingeschränkten Modifikationstechnik (vgl. Kapitel 4.2.6) durchgeführt werden.

4.2.3 Objekterzeugung basierend auf existierenden Objekten

Die Erzeugung von komplexen Volumenmodellen beginnt gewöhnlich mit einem Basismodell, welches ein erstes Primitiv sein kann, an das sukzessive Material angebracht oder von dem Material subtrahiert wird. In vielen Fällen ist es nicht zweckmäßig, die Volumenprimitive unabhängig von dem Basismodell im Raum zu erzeugen, anschließend zu plazieren und schließlich eine Boolsche Operation durchzuführen, um die beiden Teilmodelle zu kombinieren. Stattdessen sollten Interaktionstechniken die Möglichkeit bieten, direkt auf existierenden Objekten zu arbeiten, unabhängig davon, wie diese im Raum orientiert sind. Das im Rahmen dieser Arbeit entwickelte Verfahren zum schnellen und präzisen Picking und Snapping mit dem 3D-Cursor (siehe Kapitel 4.2.10) bildet die Basis für ein derartiges Vorgehen. Es läßt den 3D-Cursor auf die Oberfläche eines existierenden Objektes 'snappen', d.h. es zieht ihn auf den dem Cursorzentrum innerhalb des Pickradius' nächstgelegenen Punkt auf der Oberfläche und liefert u.a. die Flächennormale an diesem Punkt.

Soll nun ein Primitiv auf einem beseits existierenden Objekt erzeugt werden, so dient dessen Flächennormale der Orientierung des zu erzeugenden Primitivs; der gepickte Punkt auf der Oberfläche entspricht dem ersten Konstruktionspunkt des zu generierenden Primitivs. Die z-Achse des Primitivs wird entsprechend der Flächennormalen an dem gepickten Punkt ausgerichtet (vgl. Abbildung 39). Das Primitiv wird, anders als bei der Erzeugung im freien Raum, nicht nur transliert, sondern auch rotiert. Der erste Konstruktionspunkt bildet das Rotationszentrum.

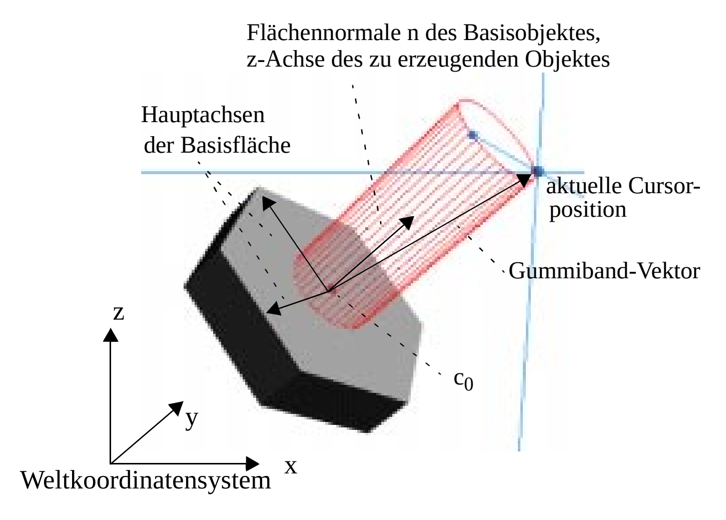


Abbildung 39: Objekterzeugung basierend auf existierendem Objekt

Sei *n* die Flächennormale am gepickten Punkt und *z* = (0, 0, 1) die auszurichtende Hauptachse des zu erzeugenden Primitivs, so ergibt sich die Rotationsachse *r* und der Rotationswinkel ρ wie folgt:

$$r = n \times z$$

$$\rho = \operatorname{acos}\left(\frac{n \cdot z}{\|n\| \cdot \|z\|}\right)$$

Aus Rotationsachse und -winkel läßt sich eine Rotationsmatrix R bilden.

$$R = \begin{bmatrix} r_x^2 + \cos\rho(1 - r_x^2) & r_x r_y(1 - \cos\rho) + r_z \sin\rho & r_z r_x(1 - \cos\rho) + r_y \sin\rho \\ r_x r_y(1 - \cos\rho) + r_z \sin\rho & r_y^2 + \cos\rho(1 - r_y^2) & r_y r_z(1 - \cos\rho) - r_x \sin\rho \\ r_z r_x(1 - \cos\rho) - r_y \sin\rho & r_y r_z(1 - \cos\rho) + r_x \sin\rho & r_z^2 + \cos\rho(1 - r_z^2) \end{bmatrix}$$

Die Transformationsmatrix M für den Zylinder aus Abbildung 39 ergibt sich zu:

$$M = T_{c_0} \cdot R \cdot T$$

$$T = \frac{1}{2} \cdot g' \qquad \text{, mit } g' = (0, 0, g_z)$$

und T_{c_0} einer Translation vom Ursprung nach c_0.

Ist das Primitiv richtig positioniert und orientiert, so verbleibt als ein Freiheitsgrad die Rotation um die Flächennormale. Um den Benutzer von weiteren Interaktionen zu entbinden, wird das Primitiv standardmäßig entsprechend der Hauptachsen der gepickten Fläche orientiert

Für Würfel und Quader, bei denen der erste Konstruktionspunkt einen Eckpunkt definiert, kann aus der Flächennormale und den Richtungen der Hauptachsen der Basisfläche eine Orientierung abgeleitet werden. Wird anstatt einer Fläche eine Kante oder Ecke gepickt, so verbleiben zu viele Freiheitsgrade, um eine Orientierung automatisch vornehmen zu können; das Objekt wird an dem Weltkoordinatensystem ausgerichtet und kann anschließend orientiert werden.

4.2.4 Implizite Boolsche Operationen

Bei Verwendung eines 2D-Eingabegerätes werden Boolsche Operationen auf Anforderung des Benutzers ausgeführt und erfordern ein vorheriges Selektieren der zu verschneidenden Objekte. Die Verwendung eines 3D-Eingabegerätes ermöglicht es, Boolsche Operationen schon während der Erzeugung von 3D-Primitiven automatisch und implizit durchzuführen. Dabei geht die Art der Boolschen Operation (Addition oder Subtraktion) aus der Erzeugungsrichtung eines Primitivs hervor. Zeigt die z-Achse des in der Erzeugung befindlichen Primitives in Richtung der Flächennormale, wird eine Boolsche Addition ausgeführt, zeigt sie in die negative Normalenrichtung, eine Subtraktion (siehe Abbildung 40). Die Boolsche Operation findet zwischen dem Basismodell und

dem in der Erzeugung befindlichen Primitiv statt. Ein derartiger Automatismus reduziert die Anzahl der nötigen Benutzeraktionen.

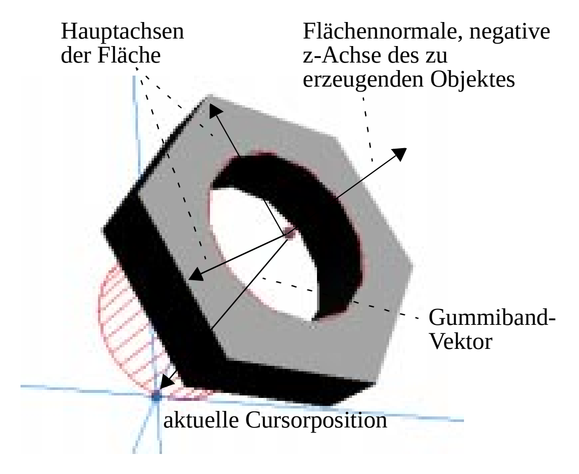


Abbildung 40: Implizite Boolsche Operationen

Da der Benutzer sich in einer direkt-manipulativen Objekterzeugung bzw. -modifikation befindet - je nachdem, ob man das Primitiv oder das resultierende Boolsche Ergebnis betrachtet - sollte auch die Berechnung der Boolschen Operation in Echtzeit erfolgen. Verfahren, um dies mit Rechnern üblicher Leistungsfähigkeit auf dem mathematischen exakten CAD-Modell durchzuführen, sind bislang noch nicht bekannt [90]. Daher müssen Verfahren eingesetzt werden, die auf einer Approximation des CAD-Modells arbeiten. Dazu zählen:

- Binary Space Partitioning Trees [106] und
- Graphik-Hardware-Puffer-basierte Ansätze [179].

Bei dem ersten Ansatz wird das facettierte Modell in einen Binary Space Partitioning Tree (BSPT) überführt, auf dem eine Boolsche Operation schneller als auf der mathematisch genauen Repräsentation des Modells ausgeführt werden kann.

Das Bildpuffer-basierte Verfahren nutzt die verschiedenen Bildpuffer (depth und stencil buffer) moderner Graphik-Hardware, um die Berechnung Boolscher Operationen durch geschicktes, mehrmaliges Rendern der dem Benutzer zu- bzw. abgewandten Objektflächen, die in die Boolsche Operation eingehen, zu simulieren. Das Verfahren ist also bildraum-basiert und arbeitet weder mit der exakten noch mit der approximierten Geometrie, sondern auf Pixelebene. Die approximierte Geometrie spielt nur insofern eine Rolle, als daß die gesamte Darstellung der Szene durch die Graphik-Hardware darauf beruht.

Im Rahmen dieser Arbeit wurden beide Verfahren implementiert. BSPTs zeigten bei komplexen CAD-Modellen numerische Instabilitäten, so daß der Ansatz nicht weiter verfolgt wurde [116]. Außerdem ist das Erstellen eines BSPTs aus einem komplexen CAD-Modell zeit- und speicherplatzaufwendig. Mit dem Bildpuffer-basierten Ansatz werden gute Ergebnisse erzielt. Die Anwendbarkeit und die visuellen Ergebnisse sind allerdings von Art und Umfang der vorhandenen Bildpuffer der gegebenen Hardware abhängig. Das (visuelle) Ergebnis der Boolschen Operation hängt von der Tiefe der Puffer ab. Bei nicht hinreichender Tiefe kann es zu Artefakten bei der Darstellung kommen. Da nur während der direkten Manipulation auf den Bildpuffer-basierten Ansatz zurückgegriffen wird, treten die Artefakte nur temporär auf. Nachdem die Interaktion abgeschlossen ist, wird die Änderung an den Modellierer propagiert, der diese auf dem exakten CAD-Modell nachzieht (vgl. Kapitel 3.3). Darüber hinaus wird aus Basismodell, erzeugtem Primitiv und Boolscher Operation ein entsprechender HistoryGraph aufgebaut.

Durch Verwendung des schnellen Bildpuffer-basierten Ansatzes zur Berechnung des Resultats einer Boolschen Operation, kann der Benutzer den Effekt dieser Operation während der interaktiven Dimensionierung eines Primitivs verfolgen. Für das Beispiel aus Abbildung 40 bedeutet dies, daß der Benutzer den Eindruck hat, in Echtzeit und direkt-manipulativ ein Loch in das Basisobjekt ’zu bohren’. Der das Loch definierende Zylinder wird als Drahtgittermodell inklusive seiner Parameter dargestellt, um dem Benutzer eine bessere visuelle Rückkopplung zu geben. Ohne eine solche Darstellung entsteht der Eindruck, daß der 3D-Cursor im Raum schwebt, ohne einen Bezug zu dem gerade erzeugten Primitiv zu besitzen.

4.2.5 Direkt-manipulatives Sweeping

Neben Boolschen Operationen können auch Extrusionen von Flächen in den Raum durch Einsatz von 3D-Eingabegeräten direkt-manipulativ durchgeführt werden. Um eine entsprechende visuelle Echtzeit-Rückkopplung zu geben, berechnet der GraphikManager aus der polygonalen Basisfläche und dem stückweise linearen Sweep-Pfad einen Polyeder. Dazu wird mit jeder Stützstelle auf dem Sweeppfad, die Menge der Ecken v der Grundfläche kopiert und mit dem durch den Sweeppfad definierten Translationsvektor transliert. Je zwei der kopierten Ecken und zwei ihrer Vorgänger definieren eine neue Mantelfläche des entstehenden Polyeders (siehe Abbildung 41). Die Deckelfläche folgt den Bewegungen des 3D-Cursors.

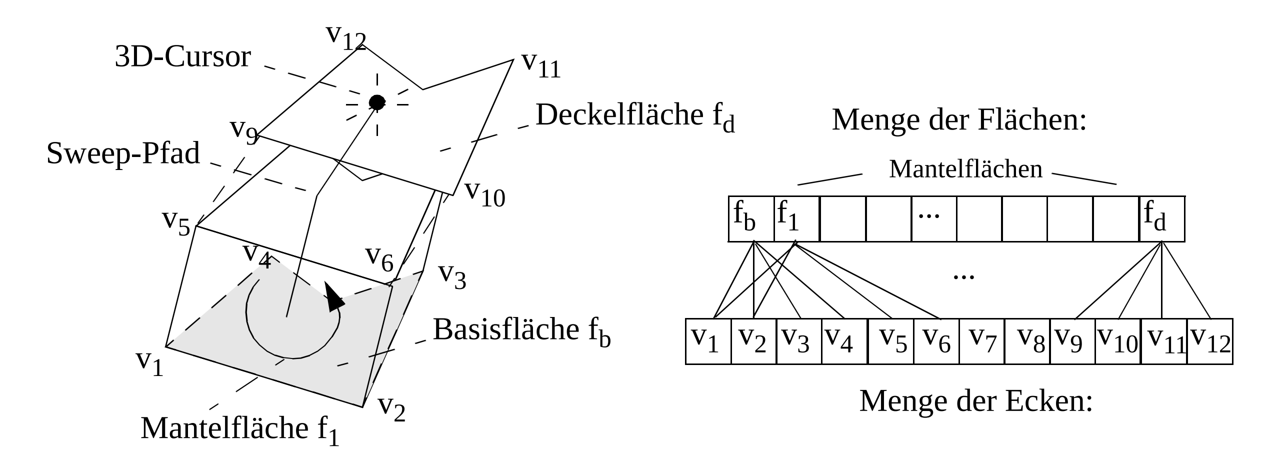
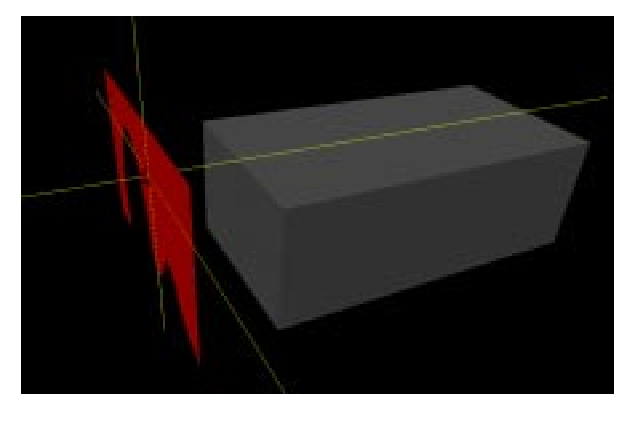


Abbildung 41: Aufbau des Echtzeit-Feedbacks beim direkt-manipulativen Sweeping

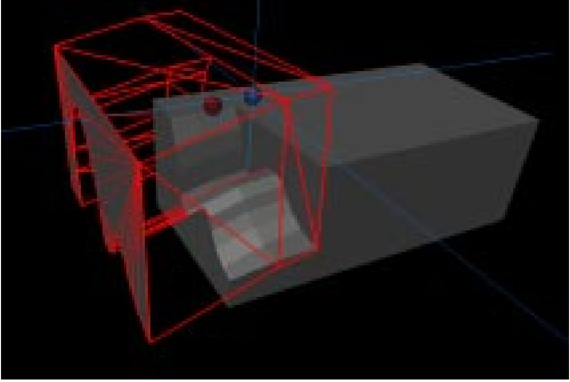
Im GraphikManager wurden Berechnungsroutinen sowohl für das rigide als auch das nicht-rigide Sweepen implementiert. Beim nicht-rigiden Sweep wird die Normale der Deckelfläche zu jedem Zeitpunkt entsprechend der Richtung des letzten linearen Abschnitts des Sweep-Pfades ausgerichtet. Die Berechnung der Position der kopierten Ecken gestaltet sich etwas aufwendiger als für den beschriebenen rigiden Fall.

Direkt-manipulatives Sweeping kann mit impliziten Boolschen Operationen kombiniert werden, z.B. zum ’subtraktiven Sweeping’. Damit lassen sich die Nachteile der Voxel-basierten Umsetzung des Ansatzes zur virtuellen Tonmodellierung (vgl. Kapitel 2.2.2.2) vermeiden. Beim subtraktiven Sweeping wird Material von einem Basiskörper (scheinbar) entfernt, während der Benutzer ein Profil (Basisfläche) durch den Raum bewegt, d.h. einen Sweep-Pfad generiert, dem die Basisfläche in Echtzeit wie beschrieben folgt. Während der Interaktion wird auf den Bildpuffer-basierten Ansatz zum schnellen Darstellen Boolscher Operationen zurückgegriffen. Nach Abschluß erfolgt die Instantiierung einer präzisen Repräsentation des Sweep-Körpers sowie

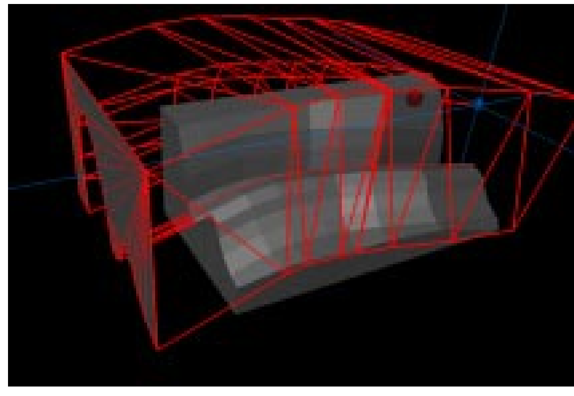
die mathematisch korrekte Berechnung der Verschneidung zwischen Basiskörper und Sweep-Körper (siehe Abbildung 42a-d).



a) Basiskörper und Sweep-Profil



b) Beginn des subtraktiven Sweepens



c) Graph. Rückkopplung während des Sweepens

d) Exaktes Resultat

Abbildung 42: Direkt-manipulatives, subtraktives Sweeping

4.2.6 Die Topologie-basierte eingeschränkte Modifikationstechnik

Bislang wurden Interaktionstechniken vorgestellt, um Primitive zu erzeugen bzw. durch die Erzeugung von Primitiven Objekte zu modifizieren, wie im Falle der impliziten Boolschen Operationen. An dieser Stelle soll mit der Topologie-basierten eingeschränkten Modifikationstechnik (topological-context based modification kurz TCBM) ein neues Interaktionsparadigma eingeführt werden, um Primitive und ihre Parameter graphisch-interaktiv zu manipulieren. Dieser neuartige Ansatz nutzt die Vorteile der 3-dimensionalen Eingabe mit ihren sechs Freiheitsgraden für Objektmodifikationen mit reduzierter Anzahl von Freiheitsgraden, wie sie im CAD typisch sind. Dazu gehören:

- Die Modifikation eines oder mehrerer Objektparameter.
- Die Rotation eines Objektes um eine Kante.
- Die Translation eines Objektes in einer Ebene.

Liang und Green [94] entwickelten den *region based reshaping* Ansatz, bei dem in Abhängigkeit von der Position des 3D-Cursors unterschiedliche Objektparameter modifiziert werden konnten. Das *region based reshaping* hat zwei gravierende Nachteile:

- Bei komplexen Objekten kommt es zu Ambiguitäten durch Überlappung von Regionen mit unterschiedlicher Bedeutung für die Manipulation.
- Es wird nicht direkt mit dem Objekt interagiert, sondern mit seiner Peripherie.

Die Topologie-basierte eingeschränkte Modifikationstechnik löst diese Probleme. Die zugrundeliegende Idee ist, Objekte in Abhängigkeit des selektierten topologischen Elementes, des geometrischen Kontextes sowie der anschließend mit dem 3D-Eingabegerät durchgeführten Bewegung (Geste) zu modifizieren. Die durch die Geste initiierte Modifikation hängt sowohl vom selektierten Objekttyp sowie den geometrischen Gegebenheiten an der selektierten Position ab. Die Modifikationsmöglichkeiten werden auf dieser Basis eingeschränkt, so daß mit sich anschließenden Cursorbewegungen nur noch Manipulationen mit einer geringe(re)n Anzahl von Freiheitsgraden möglich sind. Die Cursorbewegung wird entsprechend eingeschränkt. Zu beachten ist, daß obwohl im weiteren Verlauf der Interaktion nur Modifikationen mit wenigen Freiheitsgraden ablaufen, diese neue Interaktionstechnik die sechs Freiheitsgrade von 3D-Eingabegeräten zwingend voraussetzt und diese auf vorteilhafte Weise für Objektmanipulationen mittels natürlicher, intuitiver Gesten nutzt; ein Umschalten zwischen verschiedenen Modifikationsmodi, z.B. zwischen Translation und Rotation, entfällt. Die Gestenerkennung geht nahtlos in die direkte Manipulation über (siehe Kapitel 4.2.6.1). Das Prinzip, aus einer Geste direkt in die Direkt-Manipulation überzugehen, wurde von Rubine [125] für 2D-Gesten vorgestellt und wird hier auf 3D erweitert.

Der Begriff Geste wird hier als eine über einen gewissen Zeitraum vom Benutzer eines 3D-Eingabegerätes ausgeführte Translations- bzw. Rotationsbewegung verstanden. Bei einem tischgebundenen Eingabegerät ergibt sich die Geste aus der Summe der Relativbewegungen. Dies entspricht nicht der Bedeutung des Begriffes Geste, wie er in VR-Systemen verwendet wird, wo unter dem Begriff Geste i.d.R. eine statische Haltung der Hand verstanden wird (im Englischen beschreibt der Begriff *posture*, was sich mit Pose übersetzen läßt, diesen Sachverhalt am besten). Zur gestenbasierten Interaktion kommen dort oft aufwendige Verfahren, wie Neuronale Netze [24] oder Fuzzy Logic, zum Einsatz, um die Eigenarten unterschiedlicher Benutzer zu kompensieren.

Die Prozedur der Topologie-basierten eingeschränkten Modifikationstechnik im Überblick:

1. Selektiere ein topologisches Element
2. Führe eine 3D-Geste aus (Rotation bzw. Translation des 3D-Eingabegerätes)
3. Erkennung der Geste unter Einbeziehung des relevanten Kontexts
4. Ggfs. Darstellen eines visuellen Echos (Center Shapes), um dem Benutzer die erkannte Geste anzuzeigen
5. Einschränken der Cursorbewegung, falls notwendig
6. Ausführen der direkten Manipulation
7. Beenden der Modifikation
8. Propagieren der Änderung durch das System an den Modellierer

Die ersten drei Schritte können zur Gestenerkennung bzw. -klassifikation gerechnet werden, die im folgenden beschrieben wird.

4.2.6.1 Die Gestenklassifikationsphase

Ein direkt-manipulatives System verlangt Antwortzeiten im Bereich von einer zehntel bis zu maximal einer Sekunde, d.h. es kann nicht beliebig lange gewartet werden, bis die Bewegung, die der Benutzer mit dem 3D-Eingabegerät ausführt, als Geste erkannt wird. Auch soll die Gestenerkennung möglichst robust und stabil funktionieren und mit der Intention des Benutzers übereinstimmen. Ziel muß also die Entwicklung einer schnellen, einfachen und robusten Gestenerkennung und -klassifikation sein, die weitgehend benutzerunabhängig funktioniert.

Eine Geste wird in bezug auf das gepickte topologische Element und die lokalen geometrischen Merkmale am selektierten Punkt klassifiziert bzw. erkannt. Als lokale geometrische Merkmale gelten Normalen, Tangenten und Hauptachsen. In Kombination mit Toleranzschwellen teilen diese Merkmale den Konstruktionsraum in verschiedene Teilräume ein, anhand derer die Klassifikation der durchgeführten Geste erfolgt. Zu diesem Zweck werden die Ereignisse, die das 3D-Eingabegerät nach der Selektion eines topologischen Elementes liefert, über einen gewissen Zeitraum gesammelt und die resultierende Information bezüglich der Klassifikationsräume beurteilt. Neben den Räumen, die durch einen Toleranzwinkel gegeben sind, existiert ein Schwellwert für die Rotation, dessen Überschreiten durch die gesammelten Ereignisse zu einer Rotation des Objektes führt. Welche Interaktion nach erfolgter Gestenerkennung angestoßen wird, hängt von dem gepickten Objekttyp ab (siehe Kapitel 4.2.6.2).

Die Prozedur der Gestenerkennung kann vereinfacht wie folgt beschrieben werden:

1. Der Benutzer pickt ein topologisches Element.
2. Der Benutzer führt eine Geste aus.
3. Über einen bestimmten Zeitraum werden die durch die Geste ausgelösten Translations- und Rotations-Ereignisse gesammelt.
4. Nun wird verglichen, ob die Rotationsereignisse den Rotationsschwellwert überschritten haben und ggfs. eine Rotation eingeleitet, sonst wird mit 5. fortgefahren.
5. Mit Hilfe der Translationsereignisse wird eine neue (virtuelle) Cursorposition berechnet. Diese neue Cursorposition wird nicht visualisiert. Mit ihr wird nur festgestellt, in welchem Teilraum sich der Cursor befindet und eine entsprechende translative Modifikation eingeleitet.

<u>Selektion eines topologischen Elementes</u>

Bei der Selektion kann es sich um eine Fläche, Kante oder Ecke handeln. Abbildung 43 zeigt die topologischen Elemente und deren geometrischen Kontext, anhand dessen die Gestenerkennung durchgeführt wird. Im einzelnen sind dies an der jeweiligen gepickten Position:

- Für eine Fläche *f*: die Normale *n* und die Hauptrichtungen *u* und *v*.
- Für eine Kante *e*: die Tangente, falls es sich um eine Kurve handelt oder die Kante selbst.
- Für eine Ecke *v*: die Position der Ecke.

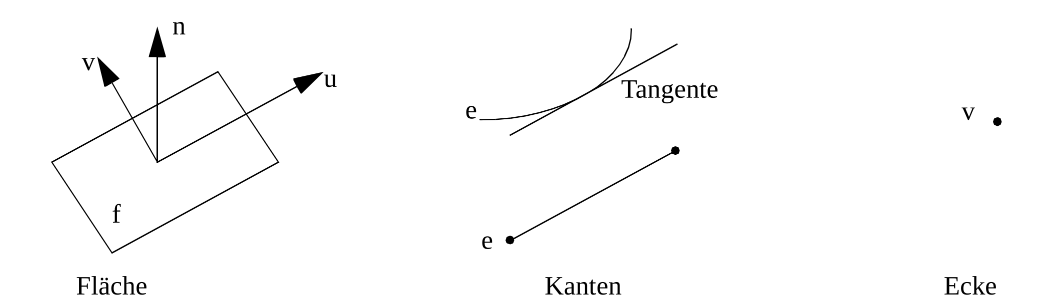


Abbildung 43: Topologische Elemente und ihr geometrischer Kontext

Anhand zweier Schwellwerte, die vom Benutzer angepaßt werden können, wird nun geprüft, ob eine Rotationsbewegung oder eine Translationsbewegung vorliegt (siehe Abbildung 44); dies läßt sich am verständlichsten am Beispiel der Kante zeigen.

Definitionen (vgl. Abbildung 44):

e sei eine Kante (*e* wird im folgenden als Vektor begriffen)

c_0 ist die Cursorposition auf der gepickten Kante zum Zeitpunkt null der Gestenerkennung

c_i ist die Cursorposition zu einem Zeitpunkt *i*

τ ist der Translationsschwellwinkel

g ist der Gestenvektor; äquivalent zum Gummibandvektor während der Objekterzeugung

ρ ist der Rotationsschwellwinkel

ϕ_i ist der zu einem Zeitpunkt *i* am 3D-Eingabegerät anliegende Rotationswinkel

r_i ist die dazugehörige Rotationsachse

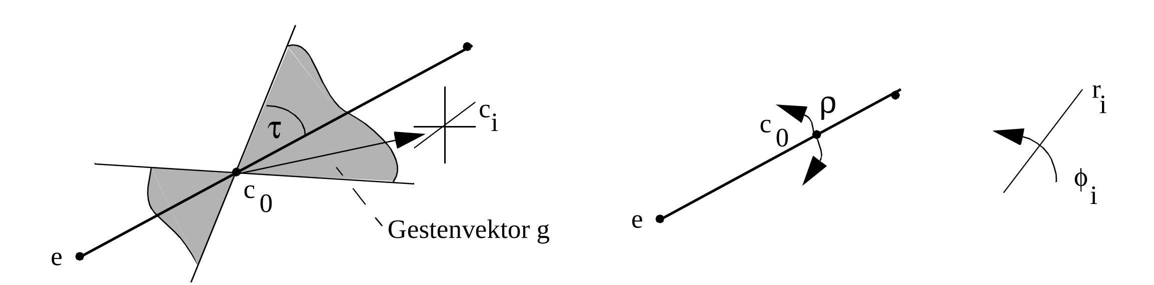


Abbildung 44: Schwellwinkel zur Klassifikation der Modifikationsgeste

In Verbindung mit *e* bildet τ einen Teilraum. Falls der Cursor nach dem Gestenerkennungszeitraum in diesem Teilraum liegt, führen alle weiteren Cursorbewegungen zur Translation des zu *e* gehörenden Objektes in die bzw. entgegen der Kantenrichtung. Ein weiterer Teilraum ist definiert durch τ und die Ebene, für die *e* die Flächennormale darstellt. Dieser Teilraum wurde aus Gründen der Übersichtlichkeit in Abbildung 44 nicht dargestellt. Die Interpretation des Teilraumes ist objekttyp-abhängig (siehe Kapitel 4.2.6.2).

Zeitintervall für die Erkennung der Geste

Zur Gewährleistung einer Systemantwort nach maximal einer Sekunde wird dieser Zeitraum als Obergrenze definiert, in dem eine Geste erkannt werden muß. Die Zeitmessung beginnt mit dem ersten für die Ermittlung der Geste relevanten Ereignis. Die Ereignisse für die Gestenerkennung werden solange gesammelt, bis eine bestimmte Anzahl von Ereignissen vorliegt oder die Zeitschranke überschritten ist (in diesem Fall werden lediglich die bisher gesammelten Ereignisse berücksichtigt, was die Erkennungsrate negativ beeinflussen kann). In Benutzertests haben sich 12 Ereignisse als das Minimum für ein mit 50 Hz arbeitendes Eingabegerät, wie z.B. der Space-Mouse, herauskristallisiert. Zwölf Ereignisse entsprechen dann in etwa einer viertel Sekunde. Im Umgang mit der SpaceMouse ungeübte Benutzer produzieren zu Beginn der Interaktion - während sie die Kappe in die gewünschte Richtung bewegen - Ereignisse, die noch nicht eindeutig auf die beabsichtigte Geste hinwirken. Aus diesem Grund werden alle innerhalb der ersten 0,2 Sekunden nach dem Start der Zeitmessung ankommenden Ereignisse von der Gestenerkennung ignoriert. So ist im Regelfall eine Systemantwort nach ca. einer halben Sekunde sichergestellt (siehe Abbildung

45). Über diesen Zeitraum *z* werden sowohl die Translations- als auch die Rotationsereignisse des 3D-Eingabegerätes erfaßt und wie nachfolgend beschrieben verarbeitet.

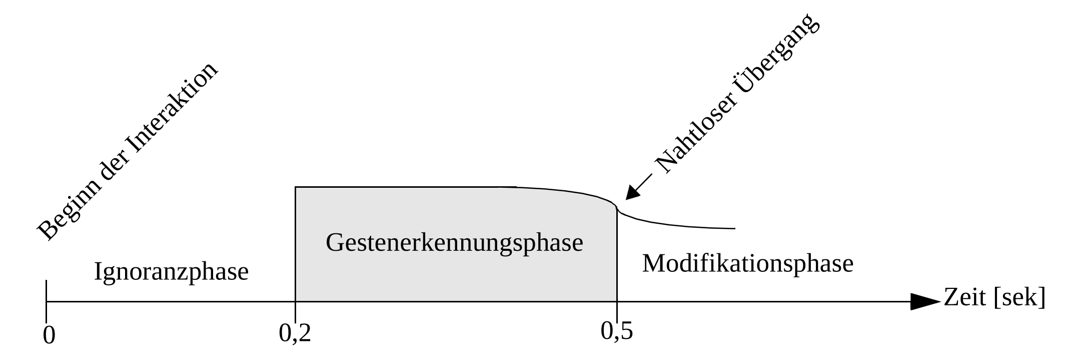


Abbildung 45: Nahtloser Übergang von Gestenerkennung nach Modifikation

Berechnung von Rotation und Translation

Die von dem 3D-Eingabegerät zu jedem Zeitpunkt *i* gelieferte Rotation kann als Rotationsachse und Rotationswinkel aufgefaßt werden. Für die Gestenerkennung sind insbesondere Rotationen um die durch den gepickten Punkt definierte Achse oder Achsen, wie z.B. Kante, Flächennormale/ -hauptrichtungen, relevant. Rotationen, deren Achse orthogonal dazu stehen, sollen keine Rolle spielen; auch hiermit wird wieder der Reiz-Reaktions-Korrespondenz Rechnung getragen.

Dies bedeutet, daß die Rotationen über den Gestenerkennungszeitraum *z* bei jedem Ereignis, d.h. zu jedem Zeitpunkt *i*, auf die Referenz-Achse abgebildet und der Rotationswinkel um diese Achse bestimmt werden muß. Um den Winkel bezüglich dieser Achse zu erhalten, wird die aus der Geste erhaltene Achse (begriffen als normierter Richtungsvektor) auf die Referenz-Achse, um die gedreht werden soll, abgebildet. Der Rotationswinkel wird mit der Länge der Projektion multipliziert. Damit wird erreicht, daß der relevante Rotationswinkel umso kleiner ist, je mehr sich der Winkel α zwischen Gestenvektor und Referenzachse 90 Grad nähert (siehe Abbildung 46).

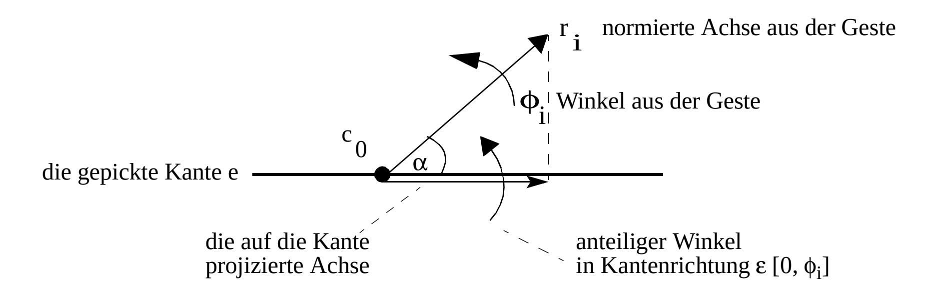


Abbildung 46: Abbildung der Rotation aus der Geste auf die Rotationsachse

Der Gesamtrotationswinkel Φ ergibt sich somit zu:

$$\Phi = \sum_{i=0}^{z} \phi_i \cdot \frac{e \cdot r_i}{\|e\| \cdot \|r_i\|}$$

Durch die beschriebene Vorgehensweise ist die *Kinästhetische Korrespondenz* [115] in 3D gewährleistet. D.h. der Benutzer muß das 3D-Eingabegerät um die Achse rotieren, um die das gepickte Objekt rotiert werden soll, genauso wie er ein reelles Objekt in seiner Hand um die Achse rotieren würde, um die er dieses drehen möchte. Somit wird eine Reiz-Reaktions-Korrespondenz auch für die Rotation hergestellt.

Die Berechnung der Translation, die der 3D-Cursor über den Gestenerkennungszeitraum durchführt, ist vergleichsweise trivial. Es sei nochmals darauf hingewiesen, daß der 3D-Cursor während dieses Zeitraums seine Position nur virtuell und nicht visuell ändert.

Der Gestenvektor ergibt sich äquivalent zum Gummibandvektor als:

$$g = c_z - c_0$$

Klassifikation der Geste

Die Entscheidung, ob der Benutzer eine Rotationsgeste oder eine Translationsgeste durchgeführt hat, erfolgt schließlich durch Vergleich mit den Schwellwerten, wobei die Rotation Priorität gegenüber der Translation besitzt. Dies hat sich als sinnvoll erwiesen, da mit einer Rotation eines 3D-Eingabegerätes auch immer eine Translation einhergeht. Die motorischen Fähigkeiten des Menschen lassen es nicht zu, eine Rotation alleinig auszuführen; eine - wenn auch geringe - translative Bewegung ist immer enthalten. Die mechanischen und sensorischen Eigenschaften von 3D-Eingabegeräten tragen ihren Teil dazu bei, daß Rotationen nicht gänzlich ohne Translationskomponenten auszuführen sind.

Daher gilt:

Wenn $\Phi > \rho$, dann führe Rotation aus.

Wenn $\operatorname{acos}\left(\frac{g \cdot e}{\|g\| \cdot \|e\|}\right) < \tau$, dann führe Translation bzw. Parameteränderung aus.

Sonst weise Geste zurück.

Damit ist die Gestenklassifikationsphase abgeschlossen. Die zur Ermittlung der Geste erfaßten Ereignisse werden nicht auf das Objekt angewendet, d.h. das Objekt bleibt bis zu dem jetzigen Zeitpunkt unangetastet. Um dem Benutzer die erkannte Geste anzuzeigen, wird ein visuelles Echo in die Szene eingeblendet (siehe Abbildung 48). Jetzt beginnt die Modifikationsphase, in der die direkte Manipulation des gepickten Objektes gemäß der im nächsten Unterkapitel beschriebenen Gesteninterpretationsregeln erfolgt.

4.2.6.2 Die Modifikationsphase

Ist die Gestenerkennungsphase beendet, so geht die TCBM nahtlos in die Modifikationsphase über, während der Benutzer seine Geste fortführt. Beim Übergang von Gestenerkennungs- in Modifikationsphase wird gegebenenfalls ein visuelles Feedback in die Szene eingeblendet und der Cursor für alle weiteren Ereignisse in seiner Bewegungsfreiheit eingeschränkt. Während der Modifikationsphase wird nicht nur das Objekt direkt-manipulativ geändert, sondern mit jedem Ereignis auch die Position oder ggfs. das Erscheinungsbild des visuellen Feedbacks abgepaßt.

Betrachtet man verschiedene Typen von Volumenprimitiven, so wird schnell klar, daß ein und dieselbe Geste auf verschiedene Primitive angewandt, nicht zwangsläufig zu einer gleichartigen Modifikation führen kann. So sollen durch nur zwei Arten von Gesten (Translation und Rotation) folgende Modifikationen unterstützt werden:

- Änderung eines Parameters
- Gleichzeitige Änderung mehrerer Parameter (Skalierung)
- Rotation des Objektes
- Translation des Objektes.

Aus diesem Grund wird eine kontext-abhängige Interpretation der Geste benötigt, die bestmöglich mit der Intention des Benutzers übereinstimmt. Wie die Interaktionstechniken zur Erzeugung von Primitiven, müssen auch die Modifikationsregeln semantisch-korrekt arbeiten. Die Volumenprimitive sollten sich dabei so verhalten, als ob sie aus Gummi bestehen würden, wobei eine Modifikation lokal wirken kann. Beispielsweise sollte mit dem Verschieben der Deckelfläche eines Zylinders nicht zwangsläufig eine Modifikation der Bodenfläche einhergehen.

Die erwartungs-konforme Interpretation der Modifikationsgesten wurde anhand einer prototypischen Umsetzung für die Primitive Quader und Zylinder experimentell ermittelt. Die verallgemeinerten Modifikationsregeln für die verschiedenen Volumenprimitive sind in Tabelle 4 zusammenfassend dargestellt. Die erste Spalte enthält das gepickte topologische Element. In Spalte 2 ist die Geste in bezug zu dem jeweiligen Objekttyp im Modellkoordinatensystem (MKS) des Objektes dargestellt. In der dritten Spalte ist die Transformationsregel und in der vierten Spalte etwaige Ausnahmen von dieser Regel aufgeführt. Zu beachten ist, daß in Tabelle 4 alle translativen Gesten, die zur Änderung eines oder mehrerer Parameter eines Objektes führen, als Skalierung ausgedrückt werden. Diese Darstellungsform wurde zur Vereinfachung der mathematischen Beschreibung gewählt. Tatsächlich werden solche translativen Gesten auf Parameteränderungen der Volumenprimitive abgebildet.

Definitionen und Abkürzungen:

s	=	Skalierungsvektor vor der Modifikation
m	=	Modifikationsvektor; ergibt sich aus dem Gestenvektor nach Anwendung der Bewegungseinschränkung für den Cursor und Projektion ins MKS; in Tabelle 4 ist m als Pfeil dargestellt
n	=	Normalenvektor
opp	=	*opposite picked point*: Der Punkt, gegeben durch den doppelten Vektor von der Pickposition zum Zentrum des Primitives
cobf	=	*center of bottom face*: Zentrum der unteren Fläche
copf	=	*center of opposite picked face*: Zentrum der Fläche, die der planaren Fläche gegenüberliegt, welche zur gepickten Kante adjazent ist
Ursprung	=	Nullpunkt für die Skalierung
$\lvert v\rvert$	=	Vektor v normiert
$\lVert v\rVert$	=	Länge/Betrag eines Vektors v

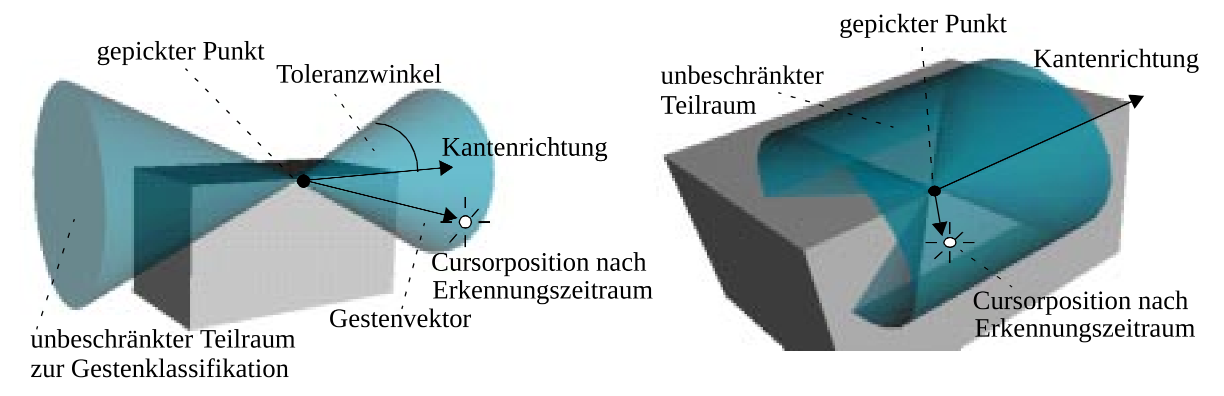
Tabelle 4: Modifikationsregeln der TCBM

Topologisches Element	Geste	Transformationsregel	Ausnahme
Fläche	orthogonal zu der Normalen	transliere Objekt in der Ebene	
planare Fläche	in Richtung der Normalen	$s' = s + m$ Ursprung = *opp*	
nicht-planare Fläche	in Richtung der Normalen mit $n_z = 0$	$s' = s + m$ Ursprung = *cobf*	Kugel: Cursor beschränkt auf *n* $s'_i = s_i + ((\|m\| * n) * \|\|m\|\|)$, $i \in x,y,z$ Ursprung = Zentrum der Kugel
Ecke	Rotation	rotiere Objekt um die Ecke	
	Translation	$s' = s + m$ Ursprung = *opp*	Konus: Cursor beschränkt auf die z-Achse
Kante	in Richtung der Kante/Tangente	transliere Objekt entlang der Linie	
	Rotation um die Kante	rotiere Objekt um die Kante/Tangente	
kreisförmige Kante	in Richtung des Krümmungsvektors	$s'_i = s_i + ((\|m\| * n) * \|\|m\|\|)$, $i \in x, y$ Ursprung = *cobf*	
	orthogonal zum Krümmungsvektor	$s'_z = s_z + m_z$ $s'_{xy} = s_{xy} + ((\|m_{xy}\| * n_{xy}) * \|\|m_{xy}\|\|)$ Ursprung = *copf*	
elliptische Kante	in Richtung des Krümmungsvektors	$s' = s + m$ Ursprung = *cobf*	
	orthogonal zum Krümmungsvektor	$s' = s + m$ Ursprung = *copf*	

Prismatische Objektformen können wie die entsprechenden konischen Formen behandelt werden. Horizontale Kanten im Modellkoordinatensystem müssen dabei von vertikalen unterschieden werden, um eine erwartungskonforme Interpretation zu erreichen; ebenso Mantel-, Boden- und Deckelflächen.

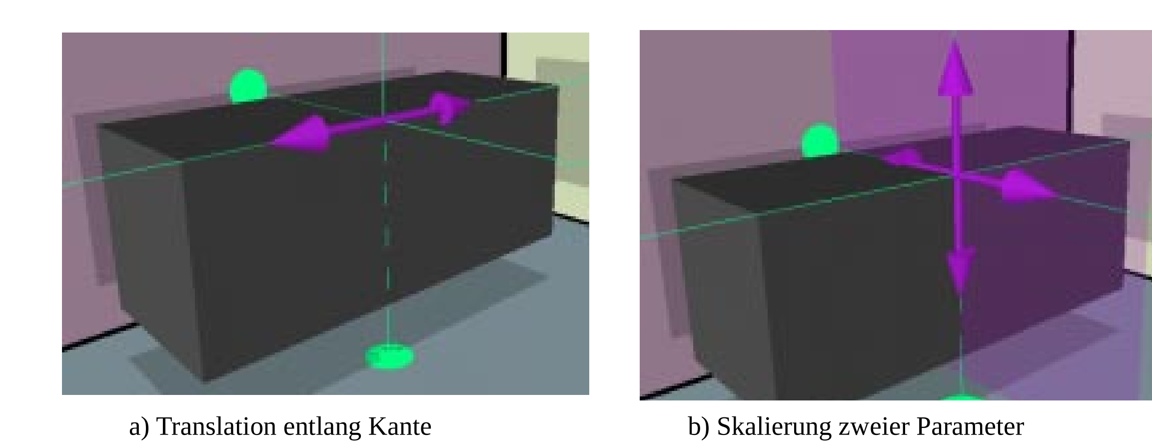
Ein Beispiel soll die Funktionsweise der TCBM verdeutlichen: Abbildung 47 zeigt, wie ein Quader durch verschiedene Gesten nach dem Picken derselben Position auf unterschiedliche Weise modifiziert werden kann. In Abbildung 47a) wird der 3D-Cursor nach der Selektion der Kante in Richtung dieser Kante bewegt. Der Cursor liegt bei Ablauf des Gestenerkennungszeitraums in dem Teilraum, der durch die Kantenrichtung und den Translationsschwellwinkel gegeben ist (in Abbildung 47 durch ein semi-transparentes Hüllvolumen visualisiert[2]). Nun wird ein

visuelles Echo eingeblendet (vgl. Abbildung 48a), das dem Benutzer die erkannte Modifikationsabsicht anzeigt. Alle folgenden Interaktionen werden auf die durch die Kanten definierte Grade abgebildet und eingeschränkt. Der Quader kann nur entlang der Geraden bewegt werden. In Abbildung 47b) befindet sich der Cursor nach dem Gestenerkennungszeitraum in dem Teilraum, der durch die Fläche orthogonal zur Kante und dem Translationsschwellwinkel definiert wird (ebenfalls als semitransparenter Teilraum visualisiert). Nun wird das in Abbildung 48b dargestellte visuelle Feedback eingeblendet. Alle folgenden Interaktionen werden auf die Fläche orthogonal zur Kante, die durch den gepickten Punkt verläuft, abgebildet. Der Quader kann nun in zwei Richtungen skaliert werden.



a) Verschiebung entlang einer Geraden b) Skalierung eines Quaders

Abbildung 47: Beispiel zur Topologie-basierten eingeschränkten Modifikationstechnik



a) Translation entlang Kante b) Skalierung zweier Parameter

Abbildung 48: Visuelles Feedback während der TCBM

Das visuelle Feedback besteht im Falle der Translation entlang einer Geraden aus einem Doppelpfeil (siehe Abbildung 48a), so daß der Benutzer erkennt, daß er den Quader in beide Richtungen verschieben darf. Im Falle der Skalierung in zwei Dimensionen besteht das graphische Echo aus einer transparenten Fläche, die orthogonal zur Kanten an der gepickten Position steht. Zwei bidirektionale Pfeile zeigen die Freiheitsgrade an (siehe Abbildung 48b).

Es sei nochmals darauf hingewiesen, daß die Änderungen während der Modifikationsphase alleine vom GraphikManager durchgeführt, d.h. berechnet und visualisiert werden. Nach Abschluß der Modifikationsphase wird die Änderung auf die mathematisch korrekte Objektrepräsentation im

2. Dieser Raum ist in Richtung der Bodenfläche, der zur Visualisierung verwendeten Kegel, offen.

Modellierer übertragen (vgl. Kapitel 3.3). Dadurch ist eine direkte Manipulation in Echtzeit möglich. Wie bei der Objekterzeugung, kann die Interaktion auch bei der Modifikation diskretisiert werden, zum präzise Manipulaionen durchführen zu können (siehe Kapitel 4.2.9).

4.2.7 Modifikation komplexer Modelle

Im vorangegangenen Unterkapitel wurde die Modifikation einzelner Volumenprimitive mit Hilfe der Topologie-basierten eingeschränkten Modifikationstechnik (TCBM) dargestellt. Im Konstruktionsprozeß müssen oft komplexe, zusammengesetzte Modelle modifiziert werden. Dies leistet die TCBM in Verbindung mit dem HistoryGraphen und dem sog. *HistoryPick*. Dabei wird die Fähigkeit des HistoryManagers genutzt, Veränderungen in der Historie von Objekten automatisch 'vorwärts' zu propagieren, so daß das Resultat aktualisiert wird.

Beim HistoryPick wird die 3D-Pickposition rekursiv 'rückwärts' durch den HistoryGraphen propagiert, um das gepickte Primitiv innerhalb eines komplexen Modells zu identifizieren. Die Transformationen, die das Modell seit der Erzeugung dieses Primitivs durchlaufen hat, werden aufsummiert. Um dem Benutzer die Modifikation darzustellen, wird eine Kopie dieses Primitivs an der Stelle erzeugt, die der Benutzer gepickt hat. Die Kopie wird mittels der TCBM direktmanipulativ geändert. Nach Abschluß der Modifikation wird die Veränderung auf das Original des Primitivs angewendet, die Kopie gelöscht und der HistoryManager sorgt dafür, daß das Ergebnis aktualisiert wird. Damit ist eine Quasi-Direkt-Manipulation von komplexen Objekten auf Basis der Konstruktionshistorie möglich.

Um den Algorithmus darzustellen, sei die Tatsache, daß jedes Objekt im HistoryGraphen drei Transformationen beinhaltet, nochmals ins Gedächtnis zurückgerufen. Zum Auffinden des gepickten Primitivs spielt in der Rekursion nur die Basistransformation t_b und die aktuelle Transformation t_c sowie die Differenztransformation zwischen diesen beiden eine Rolle. Die Differenztransformation für ein beliebiges Objekt ist:

$$\Delta t = t_c \cdot t_b^{-1} \quad .$$

Um die Kopie des gefundenen Primitivs später richtig positionieren zu können, muß beim Abstieg durch den HistoryGraphen in jedem Schritt die Delta-Transformation Δt des aktuellen Objektes zu der des vorherigen aufmultipliziert werden. Somit entsteht als Produkt der Differenz-Transformationen für ein Primitiv p:

$$\Delta T^p = \prod \Delta t_i \qquad \text{, für alle Objekte i, die auf dem Weg zu p traversiert werden}$$

Weiterhin wird die Cursorposition in das jeweilige Modellkoordinatensystem des HistoryObjektes transformiert, um dort einen 3D-Pick (siehe Kapitel 4.2.10) ausführen zu können. Mit jedem Rekursionsschritt wird die Differenz-Transformation ΔT_i sowie die transformierte Cursorposition in einer *HistoryPickList* gespeichert, so daß nach Rückkehr aus der Rekursion entsprechende Differenz-Transformationen und Cursorpositionen für alle untersuchten HistoryObjekte vorliegen. Sobald das nächstgelegene Primitiv feststeht, kann die dazugehörige Differenz-Transformation aus der *HistoryPickList* benutzt werden, um die zu modifizierende Kopie des gefundenen Primitivs zu erzeugen und so zu positionieren, daß es an der gepickten Position des komplexen

Objektes erscheint. Die anschließende Modifikation gemäß den Regeln der TCBM wirkt sich direkt auf die Kopie aus. Abschließend wird die Modifikation unter Berücksichtigung der Differenz-Transformation auf das eigentliche Primitiv angewandt und der HistoryManager sorgt für eine entsprechende Aktualisierung des komplexen Modells. Das Kopieren des Primitivs ist nötig, um ein Echtzeitverhalten und eine Quasi-Direkt-Manipulation zu erreichen, da das Propagieren der Änderungen durch den HistoryGraphen bei jedem Interaktionsschritt zu lange dauern würde.

Der Rekursionsalgorithmus ist nachstehend als Pseudo-Code angegeben:

```
handleHistoryPick (HistoryList, PickRad, aktuellesObjekt)
{
  transformiere Cursorposition auf t_c des aktuellen Objektes
  summiere Delta-Transformation
  for (alle HistoryObjekte des aktuellen Objektes)
     if (HistoryObjekt hat keine Nachfolger) // Grundprimitiv
          führe 3D-Pick auf diesem Primitiv mit transformierter Cursorposition aus
          if (Primitiv gepickt)
               if (bislang der Pick-Kugel am nächsten gelegenes Primitiv)
                    gepicktesHistoryObjekt = Primitiv
     else   // Dieses HistoryObjekt hat weitere Nachfolger
          handleHistoryPick (HistoryList, PickRad, HistoryObjekt)
            // gehe in die Rekursion
}
```

Die Modifikation eines Ergebnismodells mehrerer Boolscher Operationen und Transformationen soll an einem Beispiel veranschaulicht werden. Bei dem Beispielobjekt (siehe Abbildung 50) handelt es sich zwar um ein einfaches zur Illustration des Algorithmus' aber hinreichend komplexes Modell, das aus mehreren Primitiven zusammengesetzt ist. Abbildung 49 zeigt den Aufbau des zugehörigen HistoriyGraphen mit allen Objekttransformationen. Zwei Quader obj_a und obj_b werden mit einer Boolschen Addition verknüpft. Es entsteht das obj_c, das anschließend transformiert wird. Ein Zylinder obj_d wird von obj_c subtrahiert, so daß obj_e entsteht. Auch obj_e wird anschließend vom Benutzer transformiert.

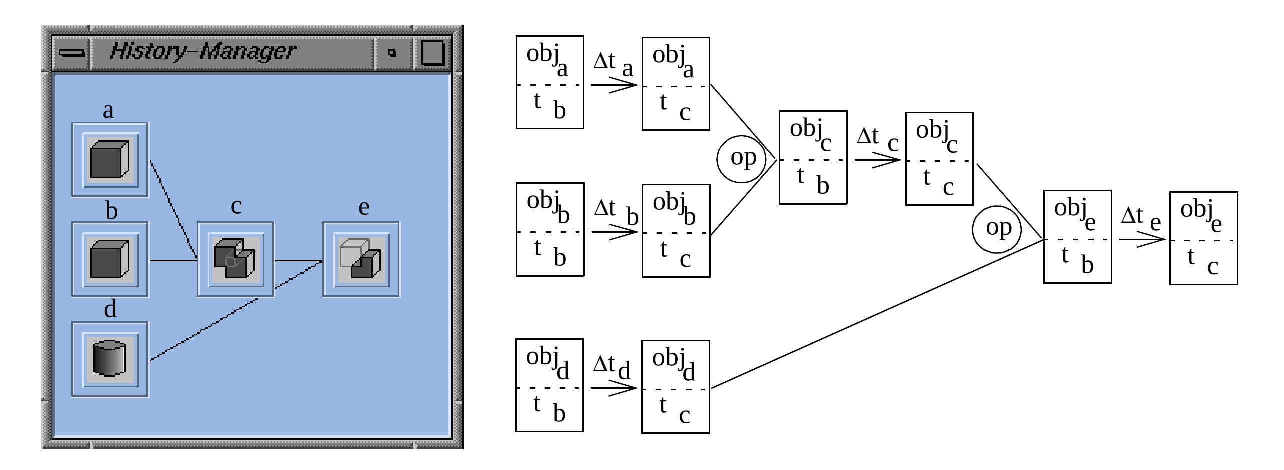
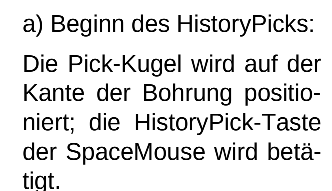


Abbildung 49: HistoryManager und History-Graph zum Beispiel aus Abbildung 50

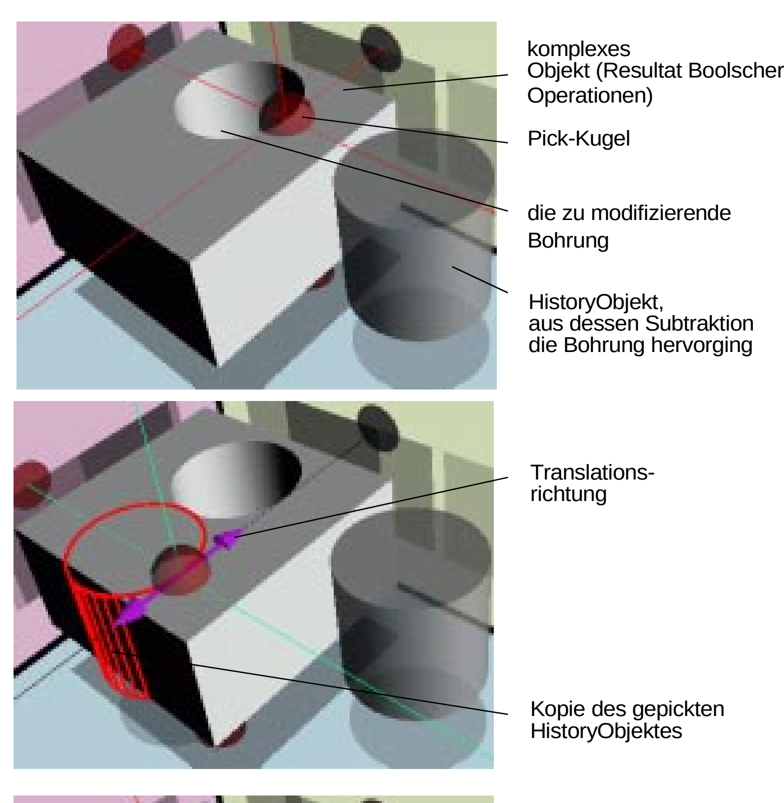
Nun führt der Benutzer einen HistoryPick auf Objekt e an der Kante des durch Objekt d hervorgerufenen Loches aus. Der Algorithmus untersucht alle ’Vorgänger’ von Objekt e im History-Graphen darauf, ob es sich bei ihnen um Blätter (Primitive) handelt; also Objekt c und d. Da Objekt c kein Primitiv (Blatt im Graphen) ist, geht die Rekursion weiter nach Objekt a und b. Ist der Rekursionsalgorithmus beispielsweise bei Objekt a angelangt, wird als aktuelle Differenz-Transformation ΔT^a in die *HistoryPickList* eingetragen:

$$\Delta T^a = \Delta t_a \cdot \Delta t_c \cdot \Delta t_e$$

Nachdem die Objekte a, b und d als Primitive bestimmt sind, wird jeweils ein 3D-Pick ausgeführt (siehe Kapitel 4.2.10). Eine Kante von Objekt d kann als nächstgelegenes topologisches Element zur ursprünglichen (transformierten) Pickposition identifiziert werden.


a) Beginn des HistoryPicks:

Die Pick-Kugel wird auf der Kante der Bohrung positioniert; die HistoryPick-Taste der SpaceMouse wird betätigt.



b) Modifikation:

Die Kopie wird am gepickten Punkt entlang der Tangente transliert. Sowohl der Ergebniskörper als auch das ursprüngliche Primitiv bleiben währenddessen unverändert; nur die Kopie (als rotes Drahtgittermodell dargestellt) wird manipuliert.

c) Abschluß:

Das HistoryObjekt wird um die auf die Kopie angewandte Differenz-Transformation verschoben, das Ergebnis wird automatisch entsprechend aktualisiert und die Kopie anschließend gelöscht.

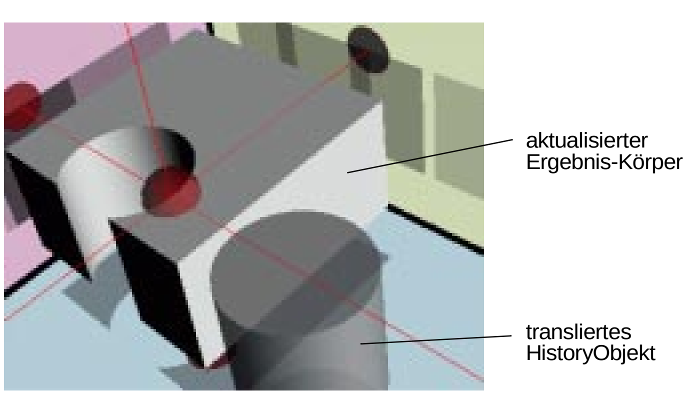
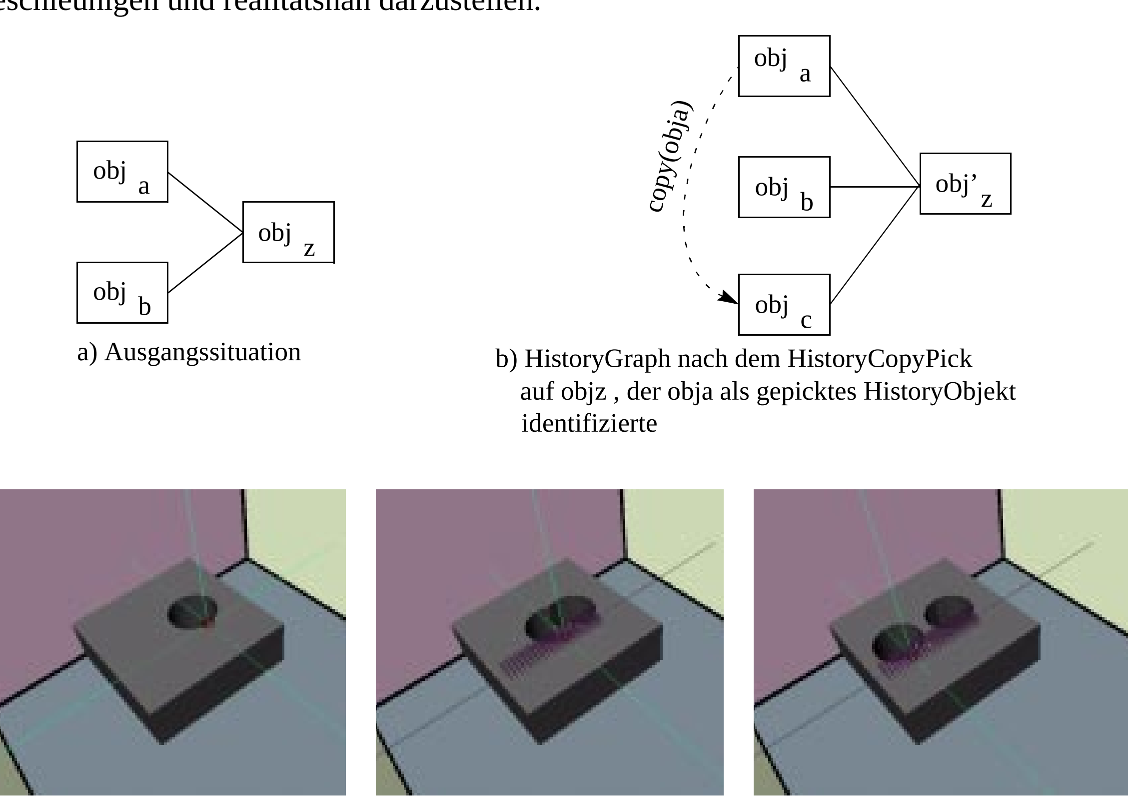


Abbildung 50: Quasi-Direkt-Manipulation komplexer Objekte [134]

Eine Kopie von Objekt d wird an der gepickten Stelle des komplexen Modells erzeugt. Die Kopie wird vom Benutzer mit einer translativen Geste verschoben und folgt der Aktion in Echtzeit. Der Translationsvektor wird nach Abschluß der Interaktion auf das Original des Primitivs angewandt,

woraufhin das komplexe Objekt automatisch aktualisiert wird. Abbildung 50 verdeutlicht den Ablauf eines HistoryPick, wobei das gepickte HistoryObjekt explizit sichtbar gemacht und transparent dargestellt wurde.

Der HistoryPick kann mit einer Kopieroperation kombiniert werden. Ein HistoryCopyPick identifiziert zunächst das gepickte HistoryObjekt, kopiert dies und fügt die Kopie mit allen Objektbeziehungen in den HistoryGraphen ein (siehe Abbildung 51a und b) - nicht zu verwechseln mit der Kopie (dem Klon) eines HistoryObjektes, die für Visualisierungszwecke benötigt wird. Die anschließende Gesten-basierte Modifikation wirkt sich auf den Klon der Kopie in der Visualisierungskomponente aus. Abschließend wird die durchgeführte Modifikation auf die Kopie angewendet und die Änderung durch den HistoryGraphen propagiert. Während der Modifikation können Boolsche Operationen implizit zwischen dem Klon und der graphischen Repräsentation des komplexen Modells durchgeführt werden. Führt der Benutzer beispielsweise einen HistoryCopyPick auf einem Loch aus und verschiebt dies mit der anschließenden Geste, so sieht er in Echtzeit, wie die Kopie des Loches durch das Material wandert und subtraktiv wirkt (siehe Abbildung 51c). Der HistoryCopyPick ist damit ein weiteres Verfahren, um Interaktionen im CAD zu beschleunigen und realitätsnah darzustellen.



a) Ausgangssituation

b) HistoryGraph nach dem HistoryCopyPick auf objz , der obja als gepicktes HistoryObjekt identifizierte

c) Ablauf des HistoryCopyPicks aus Benutzersicht: Picken des Randes der Bohrung in obj_z mit anschließender Translationsgeste; das System erzeugt eine Kopie von obj_a, die in Echtzeit mit obj_z verschnitten wird, so daß es den Anschein erweckt, als würde die Kopie der Bohrung durch obj_z bewegt.

Abbildung 51: Kopieren der Objektbeziehungen bei einem HistoryCopyPick [134]

Anzumerken bleibt, daß Modifikationen an komplexen Modellen auch durch lokale Operationen durchgeführt werden könnten. Der HistoryGraph ist aufgrund seiner Flexibilität prinzipiell in der Lage, auch solche Operationen zu erfassen und abzubilden, bei allen, bei der Kombination solcher Ansätze bekannten Probleme [89]. Diese Möglichkeit wurde allerdings nicht implementiert.

4.2.8 2-händige 3D-Interaktion

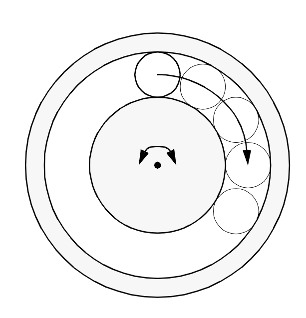
Mit der TCBM lassen sich auf intuitive Art und Weise Primitive und komplexe Modelle modifizieren, dabei werden die sechs Freiheitsgrade des 3D-Eingabegerätes auf vorteilhafte Weise genutzt, um aus der Geste des Benutzers eine Interaktion mit einer reduzierten Anzahl von Freiheitsgraden, wie sie in der Modellierung oft benötigt werden, abzuleiten. Allerdings ist die Modifikation auf ein durch das gepickte Objekt definiertes Merkmal beschränkt, so kann ein Objekt nicht um eine beliebige Kante oder in einer beliebigen Ebene gedreht bzw. verschoben werden. In Modellierungsanwendungen ist das Verschieben bzw. Rotieren eines Objektes entlang bzw. um ein topologisches Element eines anderen Objektes jedoch eine oft benötigte Funktionalität.

Eine Möglichkeit, diese Funktionalität zu unterstützen, besteht in dem sequentiellen Selektieren des Referenzmerkmals und der anschließenden Modifikation eines Objektes mit der TCBM. Wie in (Kapitel 2.2.3) dargelegt, können sequentielle Tätigkeiten durch Einführung von 2-händiger Interaktion oft parallelisiert und somit intuitiver und effizienter gestaltet werden. Eine 2-händige Erweiterung der Topologie-basierten eingeschränkten Modifikationstechnik (TCBM) unterstützt nicht nur die Objektmanipulation, sondern auch die Kamerakontrolle.

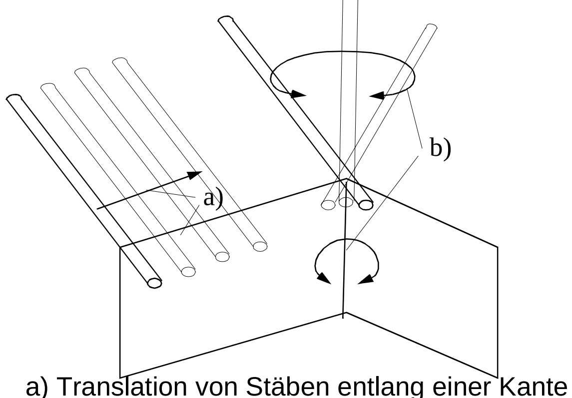
Zur 2-händigen Interaktion ist ein zweites Eingabegerät notwendig - erneut kommt ein 3D-Eingabegerät zum Einsatz. Dieses zweite Eingabegerät steuert einen weiteren 3D-Cursor. Mit diesem Cursor läßt sich ein Objekt bzw. ein topologisches Element eines Objektes picken. Weiterhin ist es möglich, das Rotationszentrum für Kamerainteraktionen mit diesem Cursor zu definieren (siehe Kapitel 4.2.8.2).

4.2.8.1 2-händige TCBM

Wie bereits erwähnt, dient die 2-händige TCBM der Translation bzw. Rotation eines Objektes relativ zu einem Referenzelement, welches zu einem anderen Objekt gehört. Zu diesem Zweck wird das Referenzelement mit der nicht-dominanten Hand und das zu modifizierende Objekt mit der dominanten Hand gepickt. Anschließend wird mit der dominanten Hand eine Geste ausgeführt und das gepickte Objekt gemäß den Regeln der TCBM modifiziert. Das visuelle Feedback wird anders als bei der einhändigen TCBM nicht an dem zu manipulierenden Objekt, sondern am Referenzobjekt eingeblendet. Während der Modifikation kann mit der nicht-dominanten Hand die Kameraposition und -orientierung kontrolliert werden, um die Szene mit Objekt und Referenzobjekt stets im Blick zu behalten.



Rotieren und Kopieren der Kugeln eines Kugellagers (Rotation um Flächennormale der mittleren Fläche).



a) Translation von Stäben entlang einer Kante
b) Rotation eines Stabes um eine Kante

Abbildung 52: Beispiele zur 2-händigen TCBM [113]

Abbildung 52 stellt beispielhaft die Möglichkeiten der 2-händigen TCBM dar. Auf der linken Seite werden die Objekte (Kugeln) sukzessive um das Zentrum des Kugellagers kopiert und rotiert. Auf der rechten Seite ist illustriert, wie ein Stab entlang einer Kante verschoben oder um eine Kante rotiert werden kann.

4.2.8.2 Kamerasteuerung mit der zweiten Hand

Die Kamerasteuerung (Navigation) ist eine in virtuellen Umgebungen essentielle Aufgabe. Man kann verschiedene Möglichkeiten zur Kamerasteuerung unterscheiden, die zwei wichtigsten und am meist verbreiteten Paradigmen sind:

- Die Kamera-in-Hand-Steuerung
- Die Szene-in-Hand-Steuerung.

Die Kamera-in-Hand-Steuerung ist besonders dann geeignet, wenn es darum geht, sich in großen virtuellen Umgebungen, wie z.B. Architekturmodellen, Schiffsmodellen, etc., zu bewegen. Hingegen bietet sich die Szene- bzw. Objekt-in-Hand-Metapher an, wenn Objekte 'von außen' inspiziert werden sollen. In der realen Welt entsprechen die beiden Metaphern dem Herumgehen um ein Objekt bzw. dem Greifen eines Objektes; durch Drehen der Hand kann das Objekt von allen Seiten betrachtet werden.

In im Rahmen dieser Arbeit durchgeführten Experimenten wurde die Kamera-in-Hand-Metapher von vielen Benutzern für Modellierungsanwendung als schlecht kontrollierbar und wenig intuitiv bewertet. Die Szene-in-Hand-Metapher wurde als intuitiver und leichter beherrschbar bewertet, allerdings wurde gefordert, das Zentrum für die Kamerarotation explizit kontrollieren zu können [113]. Das Rotieren um den Mittelpunkt oder Schwerpunkt der Szene ist nicht benutzer- bzw. aufgaben-gerecht. Zu diesem Zweck wurde an das 3D-Echo des zweiten 3D-Eingabegeräte die Möglichkeit gekoppelt, das Rotationszentrum festlegen zu können.

Obwohl diese Metapher gemeinhin als *scene-in-hand* bekannt ist, muß sie in Modellierungsanwendungen als Kamerainteraktion implementiert werden, um die Orientierung des CAD-Modells nicht permanent zu ändern, was unerwünscht wäre. Dies gilt insbesondere im Hinblick auf einen stabilen Schattenwurf der Objekte auf die Wände des virtuellen Konstruktionsraumes.

Abbildung 53 zeigt den prinzipiellen Ablauf der Szene-in-Hand-Steuerung. Zunächst werden die Translations- und Rotationswerte (t_{SM} bzw. r_{SM}) des 3D-Eingabegerätes in das Kamerakoordinatensystem abgebildet. Es ergeben sich t'_{SM} und r'_{SM}. Anschließend wird für die Kamera das Rotationszentrum in den Koordinatenursprung verschoben, d.h. von der Kameraposition KP wird der Translationsvektor t_{RZ} subtrahiert (siehe Abbildung 53c). Nun wird sowohl die Kameraposition als auch die -orientierung mit der Inversen von r'_{SM} multipliziert (siehe Abbildung 53d). Schließlich wird t_{RZ} wieder addiert und $-t'_{SM}$ angewendet (siehe Abbildung 53e).

Mit dieser Sequenz von Transformationen wird durch Manipulation der Kameraposition und Kameraorientierung der Eindruck hervorgerufen, der Benutzer hielte die Szene in der Hand.

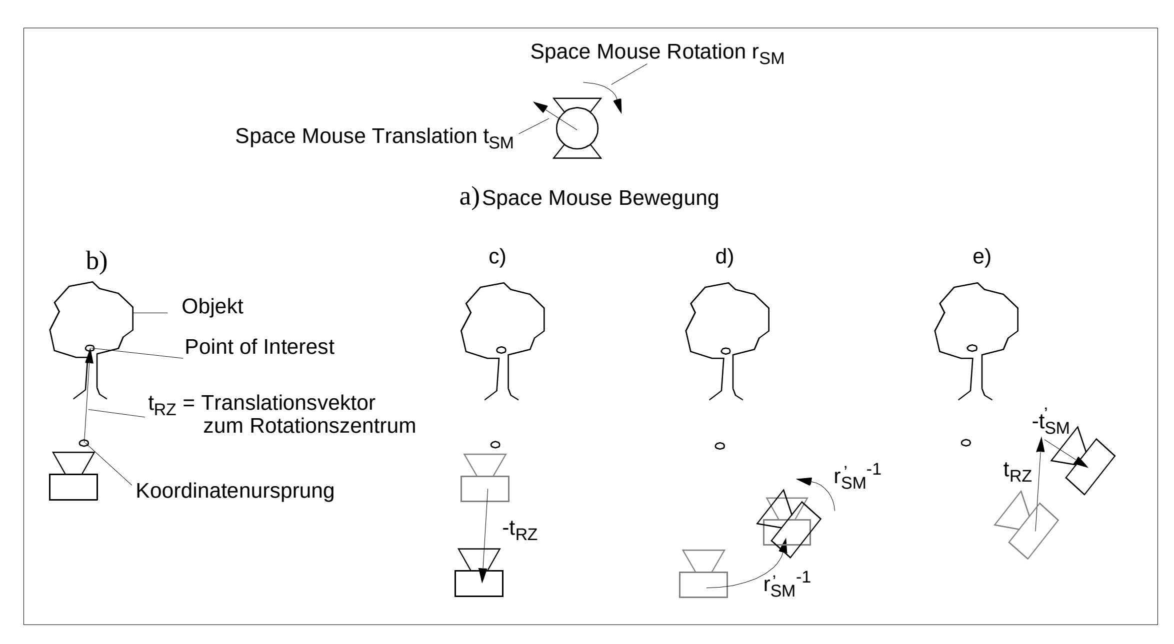


Abbildung 53: Ablauf der Szene-in-Hand-Steuerung [113]

Abbildung 54 zeigt eine Kamerainteraktion im Szene-in-Hand-Modus, wie der Betrachter sie wahrnimmt. Links oben ist die Ausgangssituation mit den beiden 3D-Cursorn zu erkennen. Links unten ist die Bewegung, die der Benutzer mit dem 3D-Eingabegerät macht, skizziert. Rechts unten ist das Ergebnis der Kamerainteraktion dargestellt.

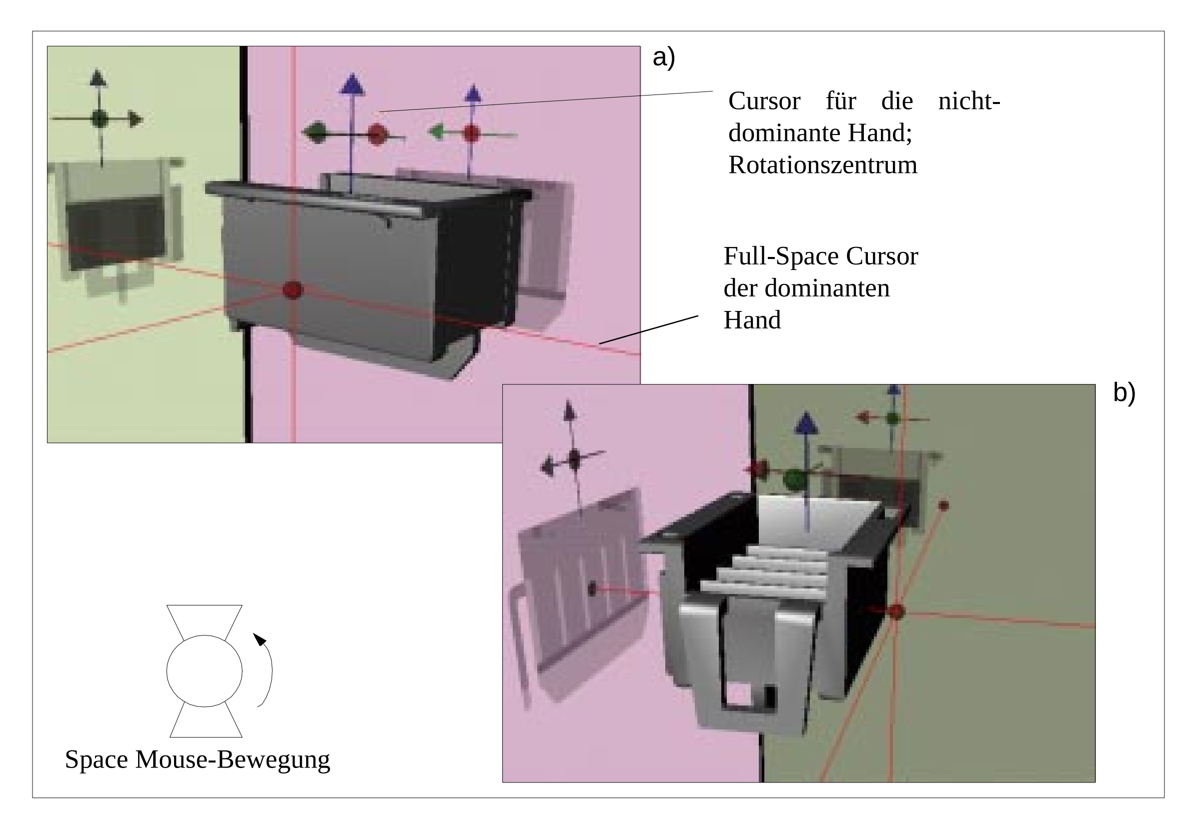


Abbildung 54: Kamerainteraktion im Scene-In-Hand-Modus [113]

An dieser Stelle sei erwähnt, daß Head-Tracking die manuelle Kontrolle der Kamera bzw. Szene nicht ersetzen kann. Der Bewegungsspielraum, den der Benutzer vor dem Bildschirm - aber auch

vor dem Virtuellen Tisch - hat, ist zu gering, um das Modell aus den gewünschten Positionen betrachten zu können. Auch der Versuch, beim Überschreiten eines Rotationsschwellwinkels mit dem getrackten Kopf in einen relativen Rotationsmodus zu gehen, um somit von der vormals abgewandten Seite auf das Modell schauen zu können, bringt keine befriedigenden Ergebnisse, da die Kamera mit dem Kopf nicht gezielt genug gesteuert werden kann.

4.2.9 Verfahren zum präzisen graphischen Interagieren

Die bislang präsentierten Interaktionstechniken dienen in erster Linie der intuitiven und effizienten Objekterzeugung und -manipulation mit 3D-Eingabegeräten. Die gesten-gesteuerte TCBM oder die Möglichkeit, Boolsche Operationen automatisch auszuführen, minimiert die Anzahl der benötigten Basisinteraktionen und steigert somit die Effizienz. Der grundlegenden Anforderung nach Präzision im CAD tragen die im folgenden beschriebenen Verfahren zum präzisen graphischen Interagieren Rechnung. Um bereits auf der Ebene graphischer Interaktionen präzise arbeiten zu können, werden im folgenden drei Ansätze eingeführt:

- die Diskretisierung,
- eingeschränkte Interaktionen und
- die parametrisierbare TCBM.

Dabei handelt es sich sowohl um neuartige Ansätze als auch um Verfahren, die aus dem 2D-Bereich bekannt sind und auf 3D erweitert bzw. verallgemeinert werden.

Wie bereits in Kapitel 2.2.2.2 erwähnt, nutzen Chu et al. [36] Sprachkommandos, um Objekte präzise zu dimensionieren bzw. zu transformieren. Auch im Rahmen der vorliegenden Arbeit wurde mit Spracherkennungssystemen gearbeitet. Insgesamt wurden drei unterschiedliche Systeme eingesetzt, um die Möglichkeiten von Spracheingabe im Modellierungsprozeß zu evaluieren. Sprachkommandos konnten zum Umschalten zwischen Dialogstati als Alternative zur Menüselektion, zur Eingabe von Parameterwerten für Objekte und zur Transformation auch in Kombination mit der gesten-basierten Modifikationstechnik (TCBM) eingesetzt werden. Die Erfahrungen lehren, daß der praktische Einsatz von Spracherkennung problematisch ist. Zwar liefern die Systeme durchaus gute Erkennungsraten, aber nur wenn wenig störende Hintergrundgeräusche auftreten. Am stärksten wirkt dem praktischen Einsatz entgegen, daß die Systeme nicht erkennen, wann der Benutzer die Spracherkennung anspricht und wann er sich mit einem Kollegen unterhält. Explizites Aktivieren der Spacherkennung durch Kommandos wie ’wake up’ oder ’listen’ wird schnell lästig; der Benutzer zieht es dann vor, auf Spracheingabe zu verzichten. Bimber [21] versucht diesem Problem mit der Verwendung eines Telefonhörers zu begegnen, in den der Benutzer sprechen muß, will er die Spracherkennung adressieren. Dieser Lösungsansatz dürfte im praktischen Betrieb bei den Mitarbeitern einer Konstruktionsabteilung ebenfalls auf eingeschränkte Akzeptanz treffen. Trotz der unbestreitbaren Fortschritte auf dem Gebiet der Spracherkennung dürfte es also noch eine Weile dauern, bis diese aus den kontrollierten Bedingungen eines Forschungslabors in die Praxis einer Konstruktionsabteilung Einzug hält.

4.2.9.1 Eingeschränkte Interaktionen

Wie bereits in Kapitel 4.1.2.4 erwähnt, werden standardmäßig die drei Translationsfreiheitsgrade des 3D-Cursors mit dem 3D-Eingabegerät kontrolliert. Manchmal ist es wünschenswert, den 3D-Cursor nur in einer Hauptebene oder entlang einer Hauptachse im Weltkoordinatensystem zu bewegen, um z.B. eine planare Fläche zu erzeugen. Zu diesem Zweck kann der 3D-Cursor in seinen Bewegungsfreiheitsgraden selektiv eingeschränkt werden.

Der 3D-Cursor kann nicht nur hinsichtlich der Hauptachsen des Weltkoordinatensystems eingeschränkt werden, sondern während der Objekterzeugung und -manipulation auch auf Hauptachsen und -ebenen des Modellkoordinatensystems des in der Erzeugung befindlichen Objektes. Dies ist besonders hilfreich, wenn ein Objekt auf einem existierenden Objekt mit beliebiger Orientierung im Weltkoordinatensystem erzeugt wird und erleichtert die selektive Manipulation einzelner Objektparameters schon während der Objekterzeugung.

4.2.9.2 Diskretisierte Interaktionen

Unter Diskretisierung wird eine Abbildung von einer Menge gegebener Werte auf eine Menge diskreter Werte verstanden. So stellt z.B. die Rundungsfunktion round(a) mit $a \in R$ eine Abbildung $\mathbf{f : R} \rightarrow Z$ dar.

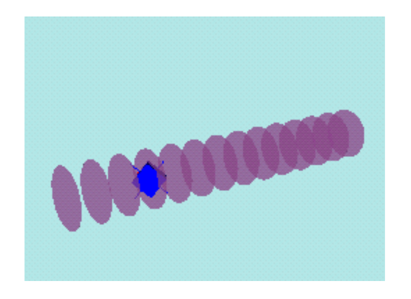
Die Diskretisierung von Interaktionen ist eine Methode, um Präzision schon im Interaktionsprozeß zu erreichen, um somit einen bestmöglichen Kompromiß aus Geschwindigkeit, Präzision und Direkt-Manipulation zu erzielen. Dies ist insbesondere in Arbeitsumgebungen, wie dem Virtuellen Tisch, wichtig, wo keine Tastatur zur herkömmlichen alpha-numerischen Werteingabe zur Verfügung steht. Alternativ kann in solchen Umgebungen zwar die Werteingabe per Sprachkommando erfolgen, was sich allerdings beim heutigen Stand der Technik der Spracherkennungssysteme als nicht praktikabel erwiesen hat.

Für den Interaktionsprozeß stellen Gitternetzlinien eine weit verbreitete Form der Diskretisierung analoger Bewegungen dar, die die Position des Mauszeigers an bestimmte diskrete Werte binden. Gitternetzlinien haben einen entscheidenden Nachteil, sie beziehen sich auf den Mauszeiger im Weltkoordinatensystem (WKS). Sind Gitternetzlinien aktiv, so können Objekte nicht mehr an beliebigen Stellen erzeugt werden. Eine Modifikation wird nicht bezüglich des bearbeitenden Objektes diskretisiert, sondern bezüglich des WKS.

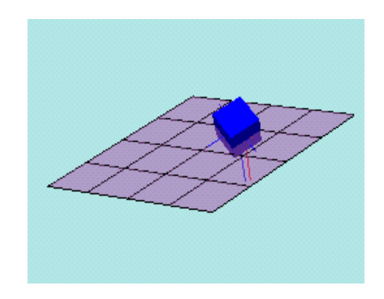
Um diese Einschränkungen aufzuheben, werden nachfolgend Möglichkeiten zur relativen (objektbezogenen) Diskretisierung von 3D-Interaktionen mit 3D-Eingabegeräten beschrieben. Die Diskretisierung kann von dem Benutzer

- für Parameteränderungen während der interaktiven Objekterzeugung und -modifikation,
- während der Translation von Objekten und
- für die Rotation von Objekten

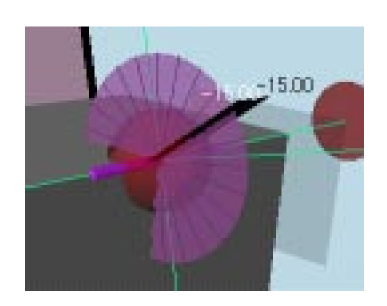
aktiviert werden. Für die drei verschiedenen Diskretisierungsarten sind unterschiedliche Diskretisierungswerte wählbar. Darüber hinaus wird dem Benutzer die Diskretisierung durch entsprechende Visualisierungshilfen dargestellt (siehe Abbildung 55).



a) Feedback für diskrete Parameteränderung



b) Feedback für diskrete Translation



c) Feedback für diskrete Rotation

Abbildung 55: Diskretisierte Interaktionen und deren visuelle Feedbacks

Die Diskretisierung der Interaktionen läuft für den Benutzer transparent ab. Die Ereignisse werden wie gewöhnlich von dem Eingabegerät entgegengenommen, jedoch nicht direkt auf das Objekt angewendet, sondern zunächst in den Parameterraum des betreffenden Objektes transformiert, dort entsprechend der eingestellten Diskretisierungswerte auf den nächstliegenden diskreten Wert abgebildet und erst dann wird das Objekt in Position, Größe oder Orientierung aktualisiert. Bei Translationen wirkt der gepickte Punkt als Nullpunkt, auf den sich die diskretisierte Interaktion bezieht. Bei Rotationen wird die aktuelle Objektorientierung als Ausgangsposition betrachtet. Das visuelle Feedback wird dementsprechend dargestellt. Bei Translationen und Parameteränderungen wird das visuelle Feedback automatisch verschoben, wenn sich der 3D-Cursor unter der Interaktion dem Rand des visuellen Feedbacks nähert; der korrekte Bezug zum Nullpunkt bleibt dabei erhalten.

Alle bislang vorgestellten Interaktionstechniken zur Objekterzeugung und -manipulation wie auch die 2-händige TCBM können diskretisiert ablaufen, womit der grundlegenden Anforderung nach Präzision im CAD Rechnung getragen wird.

4.2.9.3 Parametrisierung der TCBM

Die feature-basierte, parametrische Modellierung [26] erlaubt es dem Konstrukteur, mit Elementen und Begriffen zu arbeiten, die seiner Sprach- und Begriffswelt entlehnt sind. So sind z.B. Löcher, die in ein Modell eingebracht werden, keine Zylinder mehr, die erst erzeugt und dann mittels einer Boolschen Operation subtrahiert werden müssen, sondern per Definition subtraktiv wirkende Elemente (Features) mit weitergehender Semantik [176]. Beispielsweise kann ein Loch als Durchloch oder Sackloch ausgeprägt sein. Weiterhin kann definiert worden sein, daß ein Loch, d.h. die Mantelfläche des korrespondierenden Zylinders, mit Material umgeben sein muß. Wenn nun vom Benutzer ein Loch kreiert wird, kann aufgrund der Semantik abgeleitet werden, welche Werte für die Parameter des Lochs zulässig sind. Beispielsweise ist der maximale Radius gegeben durch den zur Mantelfläche nächstgelegenen Punkt auf der Oberfläche des umgebenden Materials. Bei einem Sackloch ist weiterhin die maximale Höhe beschränkt, da die Bodenfläche des Loches das umgebende Material nicht durchdringen darf. Hingegen ist bei einem Durchloch die Höhe so zu wählen, daß das Material komplett durchdrungen wird. Bei Feature-basierten Systemen erfolgt die Konsistenzprüfung eines erzeugten Features gegenüber seiner Umgebung i.d.R. nach der Instantiierung des Features [27]. Es ist jedoch möglich, die erlaubten Parameterwerte oder auch Transformationsbereiche für einen gegebenen Kontext aus dem Feature-basierten Modell im voraus abzuleiten [150]. Übergäbe man diese Information an die Interaktionskomponente, so könnte sie schon während der Direkt-Manipulation des Features dafür sorgen, daß das Feature semantisch-korrekt erzeugt bzw. modifiziert wird. Um den Benutzer über die Modifikationsmög-

lichkeiten in Kenntnis zu setzen, ist deren Visualisierung in geeigneter Art und Weise wünschenswert .

Es soll also möglich sein, den Bereich für Parameteränderungen, den Translationsbereich für ein Feature und die Rotationsmöglichkeiten eines solches Features einzuschränken. Bei Parameteränderungen können sich prinzipiell Abhängigkeiten untereinander ergeben. So ist in Abbildung 56a zu erkennen, daß die maximale Höhe h eines Sackloches bei gegebenem Umgebungskörper von dessen Durchmesser d abhängen kann. Für eine beliebige, aber feste Höhe h kann allerdings ein maximales d angegeben werden.

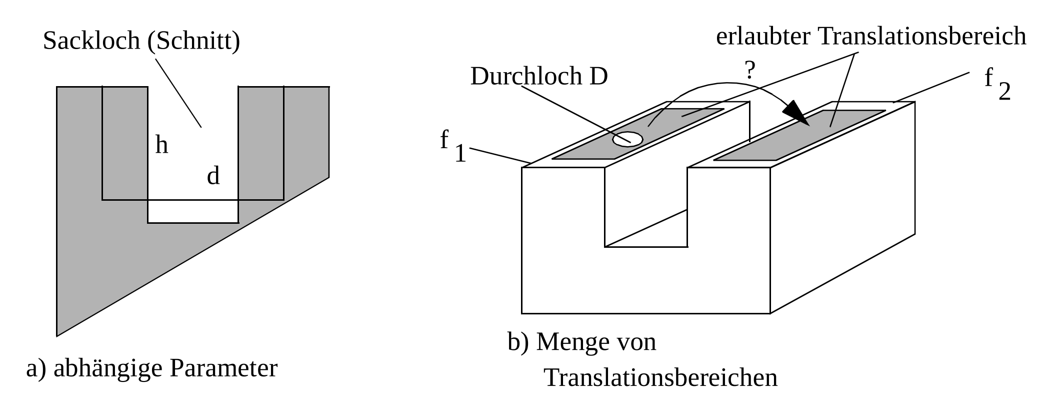


Abbildung 56: Erlaubte Parameter- und Translationsbereiche bei der Feature-Manipulation

Bei einem erlaubten Translationsbereich kann es sich generell um einen Teilraum oder gar um eine Menge von Teilräumen in R^3 handeln. Abbildung 56b zeigt eine noch recht einfache Situation: Ein Durchloch D kann auf einer planaren Fläche f_1 verschoben werden. Der Translationsbereich ergibt sich als eine um den Radius des Durchlochs D kleinere Fläche. Je nach Definition des Features ’Sackloch’ kommt auch eine Verschiebung auf die andere Seite der Nut auf Fläche f_2 in Frage. Für den allgemeinen Fall, handelt es sich bei solchen disjunkten Translationsbereichen um getrimmte Freiformflächen.

Bei einer Rotation kann sich der erlaubte Rotationsbereich ebenfalls aus einer Menge von Intervallen von Rotationswinkeln und Abhängigkeiten untereinander ergeben. Abbildung 57 zeigt ein Beispiel. Das Durchloch D kann um die Rotationsachse im Uhrzeigersinn bis a gedreht werden. Gilt für das Feature ’Durchloch’, daß die Austrittsöffnung die Fläche, die es zum Erzeugungszeitpunkt geschnitten hat, nicht verlassen darf, so kann es nur bis b gegen den Uhrzeigersinn rotiert werden. Gilt dies nicht und darf das Durchloch die Nut nicht schneiden, so darf das Durchloch weiterhin in den Bereich d-e rotiert werden. Es ist zu erkennen, daß die erlaubten Bereiche stark von der Feature-Definition abhängen. Daher ist es sinnvoll, die Berechnung der Modifikations-

möglichkeiten im Featurekern stattfinden zu lassen und parametrisierbare Interaktionstechniken anzubieten, die die Ergebnisse handhaben können.

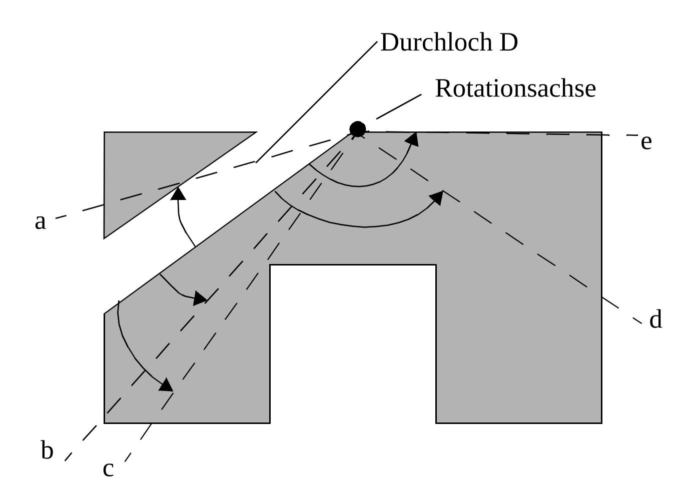


Abbildung 57: Erlaubte Rotationsbereiche bei der Feature-Manipulation

Um die Komplexität der der Berechnung der Modifikationsmöglichkeiten einzuschränken, kann zuerst die vom Benutzer beabsichtigte Modifikation ermittelt werden. Ist z.B. bekannt, ob der Benutzer ein Feature rotieren oder translieren möchte, kann ein Teil der Berechnungen, der ohne diese Kontextinformation nötig wäre, entfallen.

Mit einer Erweiterung der TCBM können erlaubte Modifikationen schon auf der Interaktionsebene gehandhabt werden. Dazu sind folgende Schritte notwendig:

- Der Entwurf einer Datenstruktur, die die aus dem Feature-basierten Modell abgeleiteten erlaubten Werte für Parameteränderungen, Translation und Rotation an die Interaktionskomponente übergibt.
- Die Erweiterung der Interaktionskomponente, damit diese die Parameter beachtet.
- Die Entwicklung geeigneter Visualisierungstechniken, die dem Benutzer anzeigen, welche Bereiche zulässig sind.

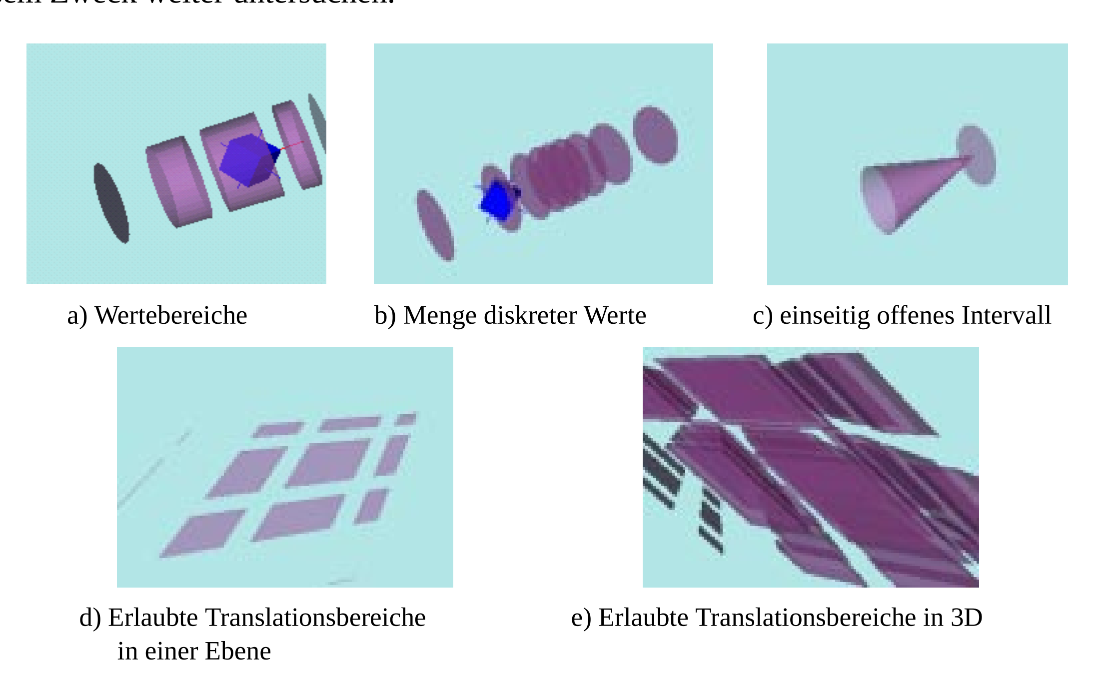
Folgende Parametrisierungsmöglichkeiten für die TCBM wurden implementiert:

- Für die Modifikation von Objektparametern:
    - Aufzählung erlaubter Werte, z.B. 1,4,5,6,9
    - Menge von erlaubten Wertebereichen, z.B. 1-3, 6-8
    - Offene Intervalle, z.B. 4 - unendlich
    - Kombinationen davon, z.B. 1, 4, 6-8, 10-unendlich
- Für die Translation von Objekten:
    - Alle obigen Wertemengen, wobei bis zu drei Parametervektoren bestimmt werden können, die nicht orthogonal zueinander stehen müssen
- Für die Rotation von Objekten:
    - Aufzählung erlaubter Werte
    - Menge von erlaubten Wertebereichen, wobei innerhalb der Wertebereiche eine Diskretisierung vorgenommen werden kann

Die Implementierung der parametrisierbaren TCBM unterstützt die Rotation um eine Achse, kann aber für beliebige Rotationen entsprechend erweitert werden. Für die Translation wird als gültiger Bereich maximal eine Menge von Parallelepipeden unterstützt. Während der Interaktion kann durch einen einfachen und schnellen Vergleich mit den erlaubten Intervallen festgestellt werden, ob die Interaktion erlaubt ist. Andernfalls kann sie ignoriert werden, so daß nur semantisch korrekte Interaktionen auf das Objekt abgebildet werden. Eine konzept-konforme Erweiterung hinsichtlich komplexerer Translationsbereiche ist ohne weiteres möglich, nur müssen in der Interaktionskomponente Einschließungsberechnungen (*point inclusion tests*) für komplexere Geometrien realisiert werden.

Die Darstellung der erlaubten Interaktionsbereiche hat sich als Herausforderung an die Visualisierung erwiesen. Für den 3-dimensionalen Fall konnte eine für den Betrachter aufschlußreiche Visualisierung im Rahmen dieser Arbeit nicht abschließend erarbeitet werden, so daß hier noch Handlungsbedarf für weitere Forschungsaktivitäten besteht.

Abbildung 58 zeigt die für die Visualisierung erlaubter Parameter- und Translationsbereiche entwickelten Darstellungsformen. Erlaubte Positionen werden durch transparente Bereiche dargestellt. Eine transparente Darstellung wurde gewählt, um das zu modifizierende Objekt nicht zu verdecken. Abbildung 58a zeigt die Visualisierung erlaubter Wertebereiche. Abbildung 58b stellt eine Aufzählung erlaubter Werte dar (man beachte, daß die Werte nicht äquidistant sind). Abbildung 58c zeigt das gewählte Visualisierungselement für ein offenes Intervall; der 'Teller' stellt die untere bzw. obere Schranke dar, der Konus gibt die Richtung an, in die modifiziert werden darf. Erlaubte Translationsbereiche in einer Ebene sind in Abbildung 58d dargestellt. Abbildung 58e verdeutlicht die Probleme, die bestehen, möchte man erlaubte 3-dimensionale Räume visualisieren. Trotz des Einsatzes von Transparenz können weder die erlaubten Bereiche, noch das Objekt hinreichend gut erkannt werden. Abhilfe können hier Visualisierungstechniken schaffen, die die Tiefeninformation, die Blickrichtung und die aktuelle Objekt- bzw. Cursorposition mit in Betracht ziehen und die einzelnen Bereiche unterschiedlich darstellen. Alternativ dazu, könnte man Techniken zur Volumenvisualisierung im Hinblick auf ihre Einsetzbarkeit zu diesem Zweck weiter untersuchen.



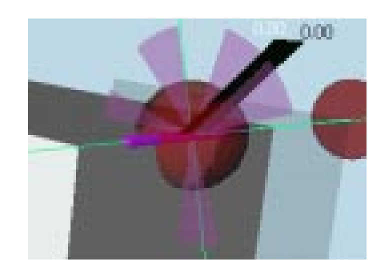
a) Wertebereiche   b) Menge diskreter Werte   c) einseitig offenes Intervall

d) Erlaubte Translationsbereiche in einer Ebene   e) Erlaubte Translationsbereiche in 3D

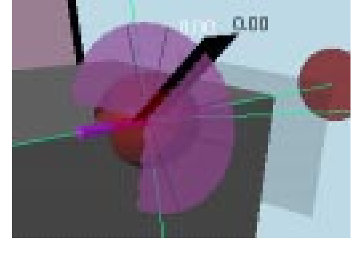
Abbildung 58: Visualisierung der erlaubten Translations- und Parameterwerte

Im Gegensatz zu den Diskretisierungsfeedbacks bewegen sich die Parametrisierungsfeedbacks nicht mit dem Cursor mit, wenn der sich ihrem Rand nähert; sie sind positionsstabil. Ist der Benutzer im Begriff, das Objekt aus einem erlaubten Bereich hinaus zu bewegen, so verharrt es an der Bereichsgrenze, um keine unerlaubte Position einzunehmen. Weitere Bewegungen des 3D-Cursors werden von der Interaktionstechnik zwar aufgefangen, aber nicht abgebildet, bis die ’virtuelle’ Position wieder in einem erlaubten Bereich zu liegen kommt. Die Bewegung des Objektes erfolgt also stets mit konstanter Geschwindigkeit, was das ’Anfahren’ einer Bereichsgrenze erleichtert, sonst würde das Objekt schlagartig in den nächsten erlaubten Bereich hüpfen, was ein Positionieren an einer Bereichsgrenze praktisch unmöglich machen würde.

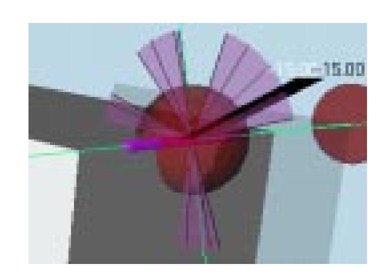
Abbildung 59 zeigt die Visualisierungselemente für parametrisierte Rotationen. Zu beachten ist, daß der aktuelle Wert sowohl graphisch als auch alpha-numerisch angezeigt wird. Für den mehrdimensionalen Fall lassen sich diese Visualisierungselemente kombinieren, dann ist es allerdings schwierig zu erkennen, welche Positionen im Raum erlaubt sind. Eine alternative Visualisierungsmöglichkeit wären Kugelsegmente, die - trotz transparenter Darstellung - wiederum schnell zu Überlappungen und Verdeckungen führen.



a) Wertebereiche



b) Aufzählung diskreter Werte



c) Diskretisierung innerhalb erlaubter Intervalle

Abbildung 59: Visualisierung erlaubter Rotationswerte

Die hier vorgestellte parametrisierbare TCBM unterstützt den Benutzer bei semantisch-korrekten Interaktion, d.h. bei der Erzeugung und Manipulation von im Sinne der Semantik validen Features und deren Position und Orientierung. Die Berechnung der Modifikationsmöglichkeiten muß aus o.g. Gründen im Featurekern erfolgen und ist nicht Bestandteil der parametrisierten TCBM. In Abhängigkeit davon, wieviel Semantik in ein Feature-basiertes Modell eingebracht wird, sind Parametrisierungen denkbar, die nicht nur die Geometrie der beteiligten Objekte in Betracht ziehen, sondern auch höherwertige semantische Informationen, wie Material, Belastung, Fertigungsgerechtheit [28]. Außer für die feature-basierte Modellierung kann die Parametrisierung auch zur Erstellung und Wahrung Normreihen-konformer Objekte [46] eingesetzt werden.

Bei der Implementierung der parametrisierbaren TCBM wurden Vereinfachungen vorgenommen, so daß nicht alle potentiell in der Realität der Feature-basierten Modellierung vorkommenden Situationen für Translation und Rotation gehandhabt werden können. Die semantisch-korrekte Änderung eines Parameters kann hingegen als komplett unterstützt betrachtet werden. In drei Punkten ist noch Handlungsbedarf gegeben:

- Bei der schnellen und exakten Berechnung der Modifikationsmöglichkeiten für komplexe Feature-basierte Modelle.
- Bei der schnellen Prüfung, ob die aktuelle Position in einem komplexen erlaubten Bereich liegt.
- Bei der Visualisierung mehr-dimensionaler Modifikationsmöglichkeiten.

Die von Fa et al. [55] vorgestellte *allowable motion inference*-Methode legt aufgrund ihres Namens Ähnlichkeit zu dem hier vorgestellten Ansatz nahe, widmet sich jedoch der Positionierung von Bauteilen bei der Zusammenbaumodellierung. Fa wertet Möglichkeiten aus, die Bewegungsfreiheitsgrade eines Bauteils einzuschränken, während das Bauteil mit einem 3D-Eingabegerät bewegt wird. Er unterscheidet folgende erlaubten Translationen und Rotationen:

- Translation in der Ebene
- Translation entlang einer Geraden
- Translation im Raum unter Beibehaltung einer Parallelitätsbedingung
- Rotation um eine Kante
- Rotation um einen Punkt

Die Vorausberechnung erlaubter Transformationen sowie Transformations- und Parameterbereiche wird von Fa et al. nicht adressiert, ebensowenig deren Visualisierung.

4.2.10 Ein Verfahren zum schnellen präzisen Picken und Snappen mit einem 3D-Cursor

Das Selektieren von Objekten und topologischen Elementen ist in graphischen Anwendungen im allgemeinen und in 3D-Modellierungssystemen im speziellen eine essentielle Aufgabe. Werden 2D-Eingabegeräte verwendet, so erfolgt die Selektion i.d.R. mit einem Pickstrahl, der vom Betrachtungsstandpunkt des Benutzers aus durch die Position des 2D-Mauszeigers geschickt wird, um das darunterliegende Element zu selektieren. Dieses als *ray pick* bezeichnete Verfahren hat sich in der Computergraphik bei Bildschirm-orientierten Anwendungen so sehr durchgesetzt, daß selbst einige Systeme, die 3D-Eingabegeräte verwenden, darauf zurückgreifen, wie z.B. JDCAD mit seinem *beam cursor*. Der *beam cursor* ist ein 3D-Cursor, der einen Pickstrahl losschicken kann, dessen Richtung vom Benutzer mittels des 3D-Eingabegerätes kontrolliert wird. Alternativ werden Objekte in virtuellen Umgebungen häufig durch Kollisionserkennung mit der virtuellen Hand selektiert ('gegriffen'). Bekannte Verfahren zur Kollisionserkennung sind nur auf triangulierten Modellen echtzeitfähig [37], [183].

Unter Snapping [15] versteht man den Effekt, daß der Cursor (oder auch ein Objekt) von bestimmten Elementen angezogen wird, wie z.B. Gitterpunkten oder anderen Objekten, wenn er sich diesen nähert. Die Elemente, auf die gesnappt werden kann, üben also eine Anziehungskraft (*gravity*) auf den Cursor aus. Snapping ist berechnungsintensiv, da mit jeder Cursorbewegung zu prüfen ist, ob der Cursor auf ein Element gesnappt werden muß. Für den 3-dimensionalen Fall trifft dies in besonderem Maße zu.

Im CAD ist das Picken und Snappen auf Basis des mathematisch präzisen Modells ein Muß. Nur so lassen sich Punkte auf der Oberfläche oder auch Informationen, wie sie die TCBM benötigt, exakt bestimmen. Das Picken bzw. das Snappen muß auf folgenden Elementen möglich sein:

- Objekten
- Flächen, Kanten, Ecken
- Mittelpunkten von Flächen und Kanten
- Extrempunkten von Kanten, z.B. bei Ellipsen

Die unterschiedlichen Elemente müssen benutzer-gesteuert kombiniert werden können und das Snapping muß schnell sein, sonst behindert es den Benutzer mehr als es ihm hilft.

Da die direkte Interaktion im Raum ein höheres Maß an Natürlichkeit und Intuitivität bietet, wurde ein Verfahren entwickelt, das echtes 3D-Picking und Snapping mit einem 3D-Cursor auf dem genauen CAD-Modell ermöglicht [158]. Durch Einführung hierarchisch-organisierter Bounding-Boxen und - im Falle des Snappings - Ausnutzung von Kohärenz zwischen den Benutzeraktionen konnte der Berechnungsaufwand soweit reduziert werden, daß das 3D-Picking und Snapping auf moderat komplexen CAD-Modellen quasi in Echtzeit ausgeführt werden kann.

4.2.10.1 Die Pickkugel - das Äquivalent des Pickstrahls in 3D

Zunächst stellt sich die Frage, wie sich der Pickstrahl aus 2D-Eingabe-basierten Anwendungen in geeigneter Weise auf 3D verallgemeinern läßt, um einerseits dem Benutzer ein einfaches Picken in 3D zu ermöglichen und andererseits keine unnötig komplexe Pick-Berechnung nach sich zu ziehen. Da ein Pickstrahl auch immer eine gewisse Ausdehnung hat, ohne die das Picken linienhafter Geometrie quasi unmöglich ist, sollte auch das Äquivalent in 3D eine Ausdehnung besitzen. Unter dem Gesichtspunkt der einfachen Abstandsberechnung und Symmetrie, wurde eine Pickkugel gewählt (die Symmetrie wird später im Zusammenhang mit der Kohärenz noch eine Rolle spielen).

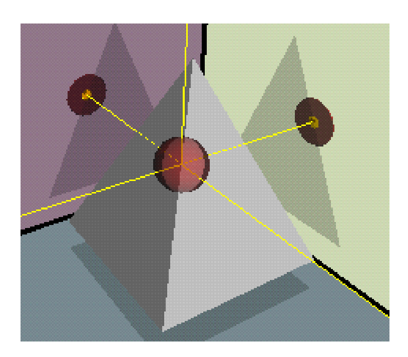


Abbildung 60: Die transparente Pickkugel

Die Pickkugel wird im Zentrum des 3D-Cursors visualisiert. Eine transparente Darstellung [185] macht die Durchdringung zwischen Kugel und Objekten leicht erkennbar (siehe Abbildung 60). Der Radius der Pickkugel entspricht dem Pickradius.

4.2.10.2 Die Pick-Heuristik

Die Pickkugel kann gleichzeitig mehrere topologische Elemente schneiden bzw. beinhalten. In einem solchen Fall, immer das nächstgelegene Element zurückzugeben, trifft oft nicht die Erwartungen des Benutzers und verhindert die Selektion von Elementen geringerer Dimensionalität (Ecken < Kanten < Flächen) und Ausdehnung. Daher wird jedem Typ von topologischem Element eine Priorität zugeordnet: Ecken haben Priorität gegenüber Kanten und Kanten haben Priorität gegenüber Flächen. Wenn die Pickkugel nun beispielsweise sowohl eine Kante schneidet als auch eine Ecke enthält, liefert die Pickprozedur die Ecke zurück, egal wie nah die Kante zum Zentrum der Pickkugel lag (siehe Abbildung 61a). Enthält die Pickkugel mehrere topologische Elemente desselben Typs, so wird das nächste zurückgeliefert (siehe Abbildung 61b). Liegt die Pickkugel im Inneren eines Objektes, so gilt das Objekt als gepickt. Da auch diese Heuristik nicht immer mit der Intention des Benutzers übereinstimmt, hat er über einen Pickfilter die Möglichkeit, den Typ des topologischen Elementes explizit zu setzen, welcher gepickt werden soll.

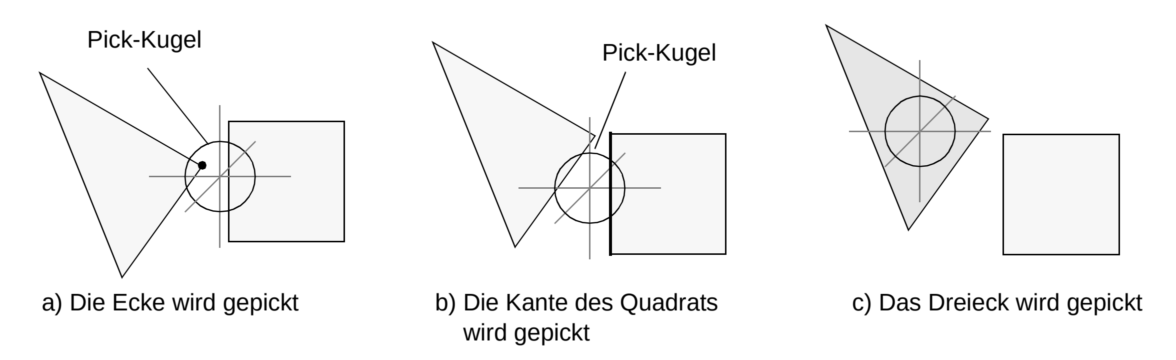


Abbildung 61: Pickprioritäten

4.2.10.3 Behandlung mehrdeutiger Situationen

Ein echter 3D-Pick läßt im Vergleich zu einer Selektion mit einem Pickstrahl eine bessere Kontrolle über das zu pickende Element zu; wo kein Pickstrahl existiert, kann es auch nicht zu Mehrfach-Selektion entlang eines solchen Strahles kommen. Trotz der Pick-Heuristik und der Möglichkeit, einen Pick-Filter zu setzen, können auch beim 3D-Pick Ambiguitäten auftreten, zu deren Auflösung weitergehende Strategien erforderlich sind.

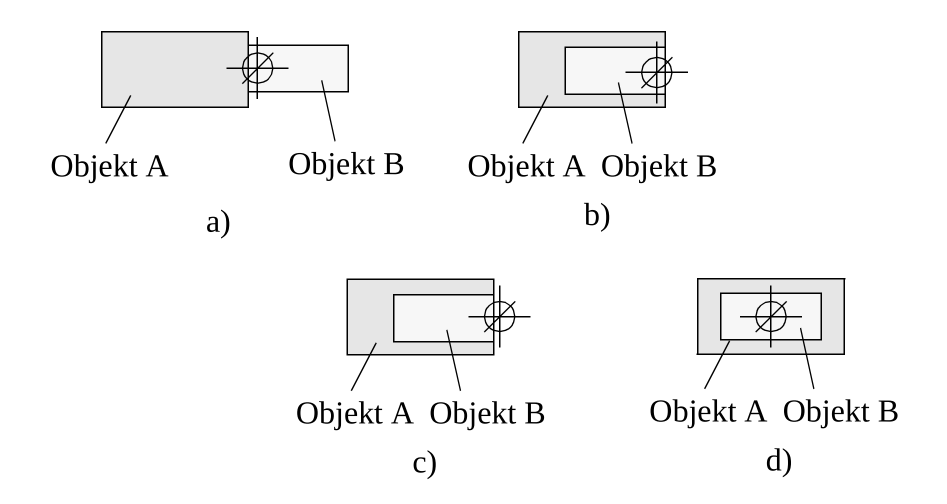


Abbildung 62: Ambiguitäten beim 3D-Pick

Abbildung 62a zeigt einen Schnitt durch zwei Objekte mit koinzidenten Flächen. Die Pickkugel schneidet beide Objekte und befindet sich in gleichem Abstand zu den beiden koinzidenten Flächen. Der Pick könnte beide Flächen zurückliefern und dem Benutzer die Entscheidung überlassen oder die Fläche, die zu dem Objekt gehört, in dem sich der 3D-Cursor befindet oder die Fläche, die zu Objekt A gehört. Als Lösungsstrategie erscheint die zweite Möglichkeit als sinnvollste.

In Abbildung 62b liegt der 3D-Cursor in beiden Objekten und die Situation bezüglich der koinzidenten Flächen ist unverändert; auch der nächste Punkt beider Objekte ist gleich weit vom Cursorzentrum entfernt. Eine Möglichkeit zur Auflösung der Doppeldeutigkeit besteht darin, zu prüfen, ob Objekt B in A enthalten ist und dann die Fläche von B zurückzuliefern. Dies ist mit einem hohen Berechnungsaufwand verbunden. Eine Alternative ist die Berechnung der Konturpunkt der gepickten Flächen mit minimalem Abstand vom gepickten Punkt. Die beiden nächstgelegenen Konturpunkte könnten denselben Abstand zum gepickten Punkt haben, so daß ein Benutzerein-

griff nötig wäre. Diese Alternative hat gegenüber der ersten Entscheidungsregel den Vorteil, daß mit ihr auch die in Abbildung 62c dargestellte Situation gelöst werden kann, bei der der Cursor außerhalb beider Objekte liegt und somit das Einschlußkriterium auf Objektebene keine Entscheidungshilfe bietet.

Die Konstellation in Abbildung 62d ist einfacher zu lösen, da die Pickkugel kein topologisches Element schneidet. Liegt die Pickkugel komplett in mehreren Objekten, so kann das Objekt selektiert werden, das den zum Cursorzentrum nächsten Oberflächenpunkt besitzt. Diese Bedingung ist oft hinreichend. Sollten mehrere solcher Objekte existieren, ist erneut eine Entscheidung durch den Benutzer erforderlich. Durch die genannte Entscheidungsregel ist es möglich, ein Objekt zu picken, das sich im Inneren eines anderen Objekt befindet. Dies ist bei Verwendung eines Pickstrahl immer mit einer zusätzlichen Benutzeraktion verbunden.

4.2.10.4 CopyPick

Das Picking kann wie auch der HistoryPick (vgl. Kapitel 4.2.7) mit einer Kopieroperation kombiniert werden. Für die TCBM ergeben sich daraus vielfältige Möglichkeiten, ein Objekt mit einem Pick zu kopieren und zu modifizieren. Diese kombinierte Interaktionsform ist insbesondere für Translationen und Rotationen interessant. So kann z.B. eine existierende Rippe in einem Arbeitsgang dupliziert und verschoben werden. Abbildung 63 zeigt ein anderes Beispiel für einen CopyPick. Hier wird ein Teil eines Bolzenhalters unter einer Rotationsgeste kopiert.

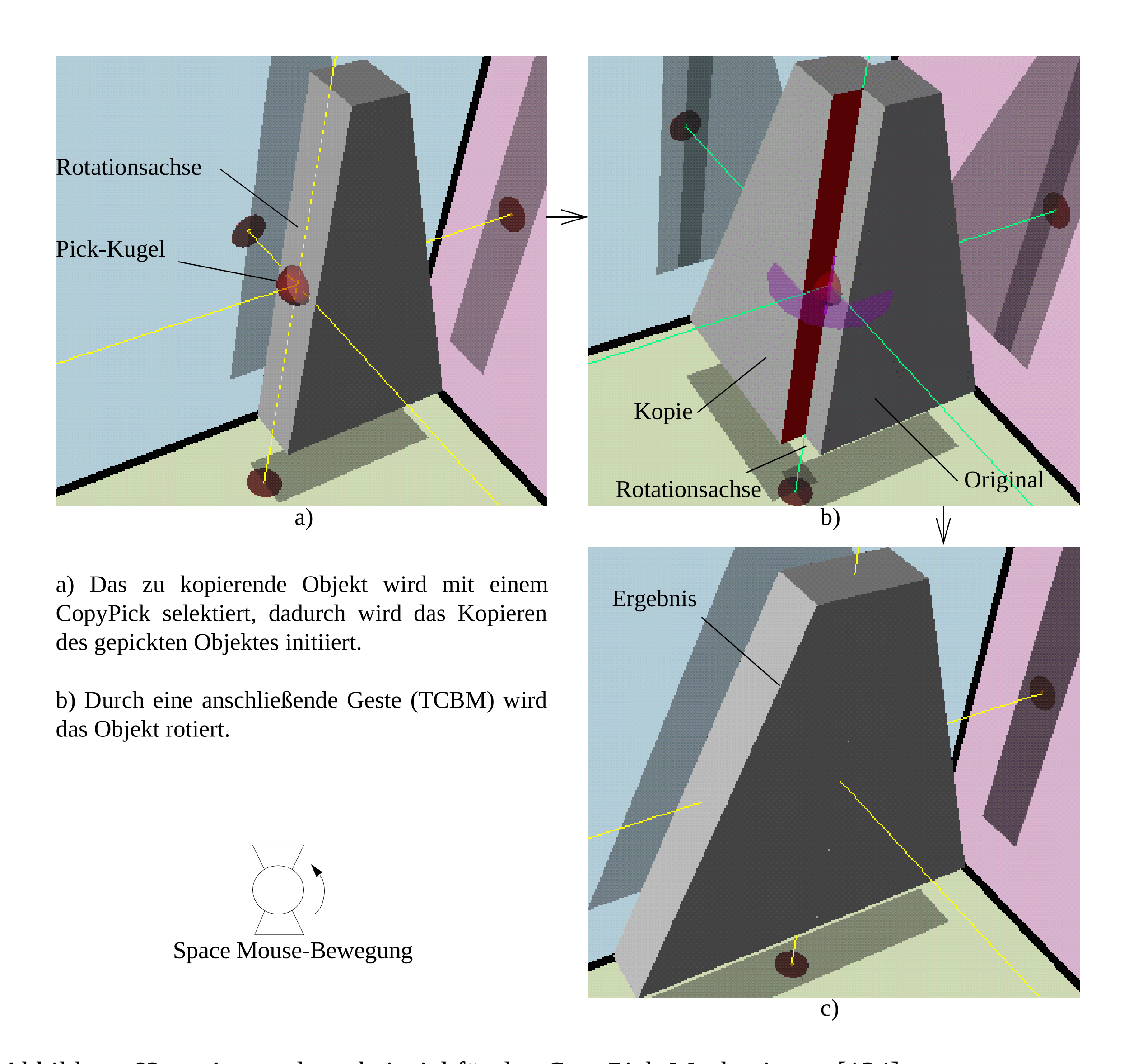


Abbildung 63: Anwendungsbeispiel für den CopyPick-Mechanismus [134]

4.2.10.5 Der Snapping-Algorithmus

Da der Picking-Algorithmus einen Sonderfall des Snapping-Algorithmus' darstellt, wird im folgenden direkt der Snapping-Algorithmus vorgestellt.

Snapping kann als sukzessives Picking mit jeder 3D-Cursor-Bewegung aufgefaßt werden. Wie bereits angedeutet, ist die Grundidee, die hinter dem Snapping - und somit auch hinter dem Picking - steht, die Berechnung des nächsten Punktes jedes Objektes zur Position des 3D-Cursors. Sofern ein oder mehrere Punkte innerhalb der Pickkugel liegen, snappt der Cursor unter Beachtung der Pick-Heuristik auf ein Objekt. Die Berechnung der nächsten Punkte jedes topologischen Elementes eines Modells zum Cursorzentrum ist zeitaufwendig. Um Snapping dennoch in interaktiven Raten ausführen zu können, ist ein Verfahren nötig, das die Anzahl dieser Berechnungen minimiert.

Anstatt den Abstand des Cursors zu jedem Objekt zu berechnen, wird zuerst die Bounding-Box aller (sichtbaren) Objekte bestimmt. Ein Objekt ist als potentieller Kandidat für das Snapping dann in die weitere Betrachtung einzubeziehen, falls ein Punkt der Pickkugel innerhalb der Bounding-

Box liegt oder anders ausgedrückt, wenn der Cursormittelpunkt in der Minkowski-Summe aus Objekt-Bounding-Box und Pickkugel liegt (siehe Abbildung 64). Da die Berechnung der Minkowski-Summe und eine Prüfung auf Enthaltensein des 3-Cursorzentrums in deren Resultat ebenfalls zeitaufwendig ist, wird als Näherungslösung auf eine in jede Richtung um den Pickradius vergrößerte Bounding-Box zurückgegriffen. Diese Näherungslösung zieht zwar mehr Bounding-Box-Checks auf der nächsten Ebene - der Ebene der Flächen - nach sich, beschleunigt das Verfahren aber entscheidend und läßt die Korrektheit des Endresultats unangetastet.

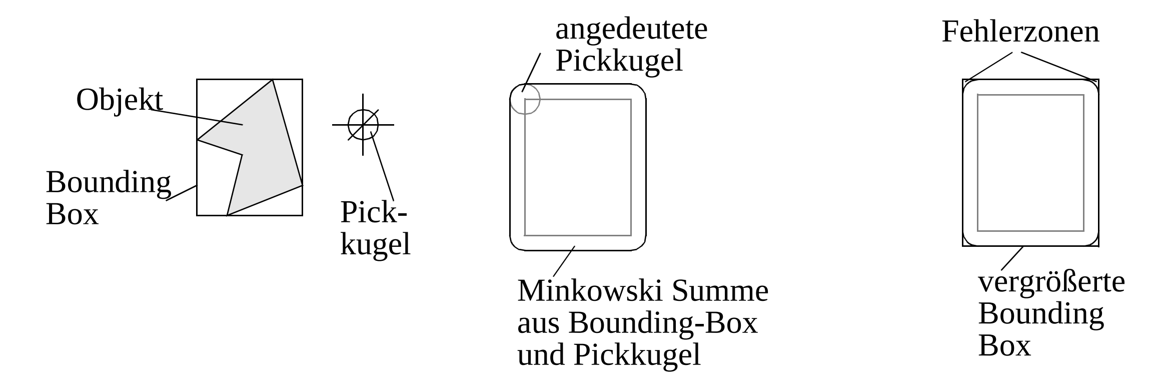


Abbildung 64: Minkowski-Summe aus Bounding-Box und Pickkugel im Vergleich zur vergrößerten Bounding-Box

Die vergrößerte Bounding-Box dient nur der Illustration, tatsächlich wird beim Vergleich der Koordinaten im Bounding-Box-Check der Pickradius miteinbezogen. Ein komponentenweiser Vergleich unter Einbeziehung des Pickradius ist im Laufzeitverhalten einer Vergrößerung der Bounding-Box mit anschließendem Komponentenvergleich überlegen. Anzumerken ist, daß sowohl die Berechnung der Minkowski-Summe als auch eine Kollisionserkennung zwischen Pickkugel und Bounding-Box aufwendinger ist, als der durchgeführte Vergleich der Cursorposition mit der expandierten Bounding-Box.

Liegt das Zentrum des Cursors innerhalb einer expandierten Objekt-Bounding-Box, werden - sofern der Pick-Filter dies erfordert - alle Flächen des Objektes daraufhin überprüft, ob das Cursorzentrum in der um den Pickradius vergrößerten Bounding-Box der Fläche liegt. Wird eine Fläche erstmalig angetastet, so erfolgt die Berechnung des nächsten Punktes auf der Fläche zum Cursormittelpunkt[3]. Die Distanz zwischen Cursor und nächstem Punkt sowie die aktuelle Cursorposition wird in einer Datenstruktur flächenbezogen abgelegt. Wird eine Fläche zum wiederholten Male angetastet, sind diese Informationen also schon vorhanden und es wird zunächst geprüft, ob der Cursor noch in der *neutralen Zone*[4] (s.u.) liegt und sich eine Neuberechnung des nächsten Punktes erübrigt. Falls nicht, findet die Berechnung des nächsten Punktes statt und die gespeicherte Cursorposition und Distanz wird aktualisiert.

Wenn der Benutzer auch Kanten und Ecken picken bzw. snappen möchte, wird das Verfahren fortgesetzt, wobei nur bei Kanten auch Bounding-Boxen zum Einsatz kommen. Bei Ecken wird die Distanz zum Cursorzentrum direkt berechnet und mit dem Pickradius verglichen. Schließlich wird geprüft, ob der Cursor im Inneren des Objektes liegt und somit das Objekt selbst und nicht eines seiner topologischen Elemente gepickt wurde.

3. Diese Funktionalität wird von dem verwendeten Modellierkern ACIS [145] bereitgestellt.
4. Durch Ausnutzung von Kohärenz zwischen Benutzeraktionen werden mit Hilfe der neutralen Zone unnötige Berechnungen vermieden.

Die neutrale Zone

Die neutrale Zone ist das Konzept, das beim Snapping durch Auswerten der Kohärenz zwischen aufeinanderfolgenden Benutzeraktionen die zeitaufwendige Neuberechnung des nächsten Punktes vermeiden hilft. Ohne neutrale Zone würde der Algorithmus die Berechnung des nächsten Punktes vornehmen, sobald sich der 3D-Cursor innerhalb einer Bounding Box bewegt. In vielen Fällen ist das Ergebnis dieser Berechnung irrelevant. Man stelle sich eine gekrümmte Fläche und ihre Bounding-Box vor (siehe Abbildung 65). Wenn der Cursor in die Bounding-Box eintritt, kann der nächste Punkt der Fläche noch um das Vielfache des Pickradius' entfernt sein. Bewegt der Benutzer den 3D-Curson in Richtung gekrümmte Fläche, so sind alle nächsten Punkt, die nicht innerhalb der Pickkugel liegen für das Snapping irrelevant; die damit verbundenen Berechnungen also unötig.

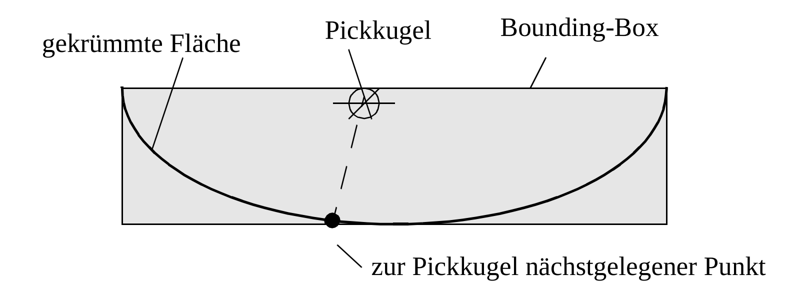


Abbildung 65: Motivation zur Einführung der neutralen Zone

Ist die Distanz zwischen der Cursorposition und dem nächsten Punkt und die Cursorposition aus einem vorangegangenen Interaktionsschritt bekannt, so kann für den aktuellen Schritt eine neutrale Zone definiert werden, in der eine Neuberechnung des nächsten Punktes nicht nötig ist (siehe Abbildung 66).

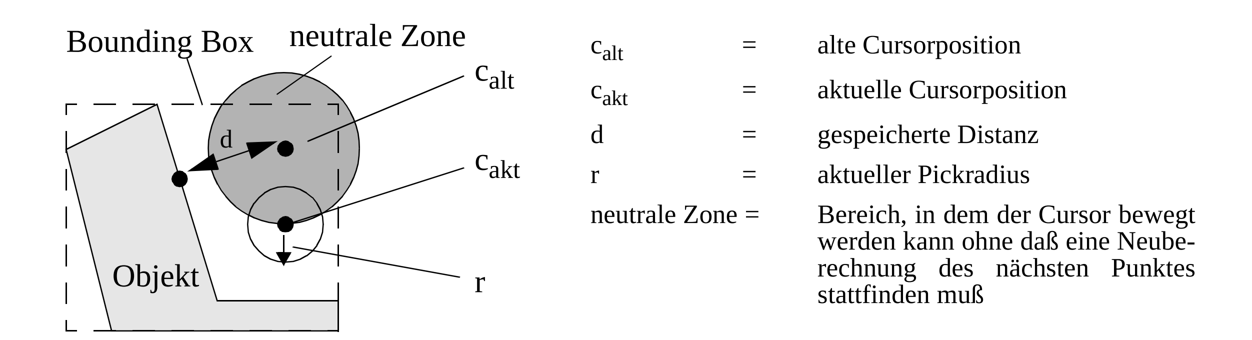


Abbildung 66: Die neutrale Zone

Die neutrale Zone sorgt gerade dann für die Vermeidung der Neuberechnung des nächsten Punktes, wenn der Cursor noch relativ weit von der Objektoberfläche entfernt und der Pickradius verhältnismäßig klein ist. Ist der Cursor weit von einer Objektoberfläche entfernt, so ist auch die Wahrscheinlichkeit größer, daß er sich noch in anderen Bounding Boxen befindet. Ist der Cursor einem topologischen Element sehr nahe, wird es immer unwahrscheinlicher, daß er sich noch in vielen anderen Bounding Boxen befindet. Im Gegenzug erhöht sich die Wahrscheinlichkeit eine Neuberechnung des nächsten Punktes durchführen zu müssen, da die neutrale Zone mit der Annäherung des Cursors an ein topologisches Element stetig kleiner wird. Damit erfüllt die

neutrale Zone den gewünschten Effekt: einerseits reduziert sie die Anzahl der Berechnungen des nächsten Punktes, wenn der 3D-Cursor noch weit vom Ziel entfernt ist und der Benutzer ihn erfahrungsgemäß schnell bewegen möchte; andererseits gewährleistet sie die Neuberechnung präziser Information, wenn der Benutzer den Cursor in der Nähe der Objektoberfläche bewegt, was erfahrungsgemäß langsamer geschieht.

Abbildung 67 zeigt beispielhaft drei unterschiedliche Vorher-Nachher-Situationen, um zu verdeutlichen, wann eine Neuberechnung des nächsten Punktes erforderlich wird: Oben links ist die Ausgangssituation dargestellt. Oben rechts hat der Cursor die neutrale Zone verlassen, befindet sich jedoch noch in der BoundingBox, so daß eine Neuberechnung des dem Cursor nächstgelegenen Punktes notwendig ist. Die Situation links unten zeigt den Cursor nach einer Bewegung innerhalb der neutralen Zone; es ist keine Neuberechnung des nächsten Punkten nötig. Rechts unten hat der Cursor sowohl die neutrale Zone (NZ) als auch die BoundingBox verlassen, auch in diesem Fall erübrigt sich die Neuberechnung.

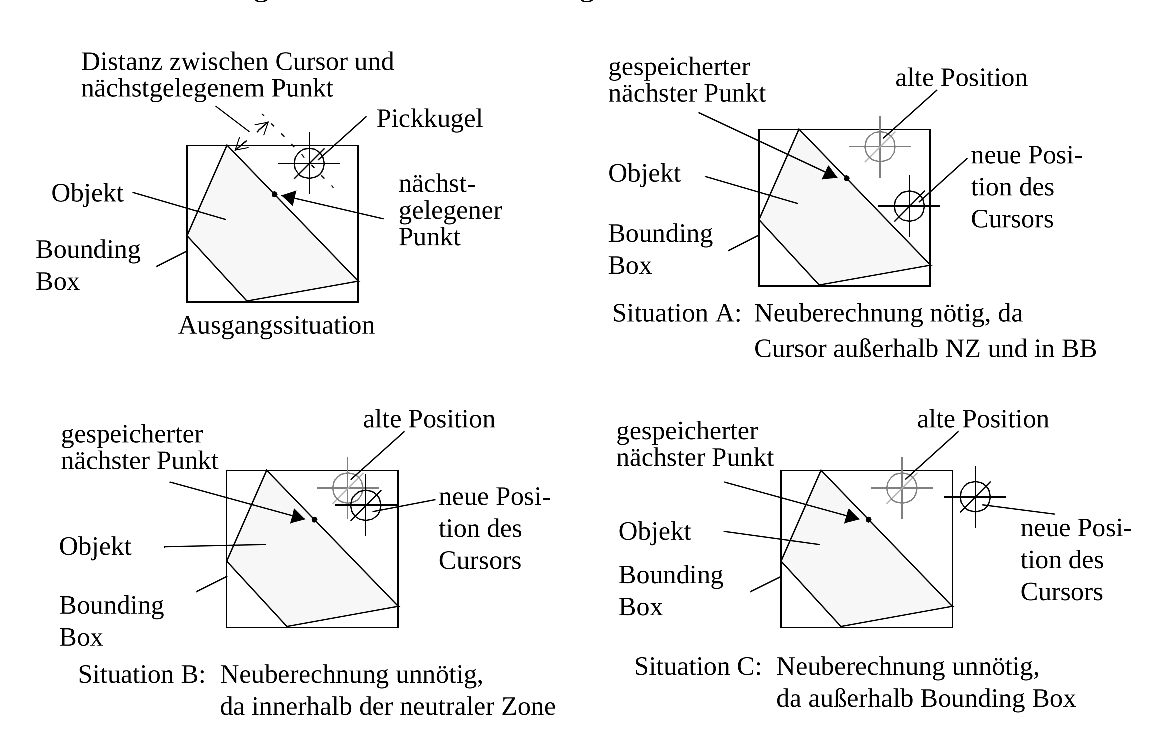


Abbildung 67: Verschiedene Situationen für die Neuberechnung des nächsten Punktes

Abschließend ist der Kern des Snapping-Algorithmus in Form von Pseudocode bis auf die Flächenebene angegeben. Der Pseudocode enthält nicht das Code-Fragment, um für jede untersuchte Fläche festzustellen, ob der berechnete nächstgelegene Punkt näher ist als der bisher nächste. Auf die Darstellung dieses einfachen Vergleichs wurde zugunsten der besseren Übersichtlichkeit verzichtet.

Der Snapping-Algorithmus in Pseudocode:

```
for (jede Bewegung des 3D-Cursors)
    for (jedes sichtbare Objekt)
        if (Cursorzentrum liegt innerhalb der erweiterten BoundingBox des Objektes)
            for (jede Fläche)
                if (Cursorzentrum liegt innerhalb der erweiterten BoundingBox der Fläche)
                    if (Fläche wird erstmalig betrachtet)
```

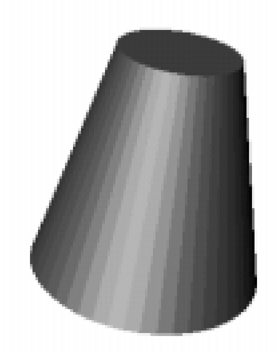
```
            {
                calcNearestPoint();    // berechne den nächsten Punkt auf der Fläche
                speichere die Distanz und Cursorposition für die betrachtete Fläche;
            }
            else
            {
              hole gespeicherte Distanz d und Cursorposition c_alt;
              if ( || c_alt - c_akt || >= d - r )  // Nearest Point muß neu berechnet werden
              {
                calcNearestPoint();    // berechne den nächsten Punkt auf der Fläche
                speichere die Distanz und Cursorposition für die betrachtete Fläche;
              }
            }
        ...
```

Anzumerken bleibt, daß das Snapping nicht nur auf den 3D-Cursor wirken kann, sondern auch auf ein mit dem 3D-Cursor selektiertes Objekt. Durch diese Kombination ist es möglich, ein gepicktes Objekt entlang der Oberfläche eines anderen Objektes zu verschieben. Das zu verschiebende Objekt wird dazu an einer Fläche gepickt und kann mit dem 3D-Eingabegerät mit sechs Freiheitsgraden durch den Raum bewegt werden; der 3D-Cursor folgt der Bewegung. Sobald der 3D-Cursor auf ein anderes Objekt snappt, wird das bewegte Objekt auf die Snap-Position verschoben und entsprechend der dortigen Flächennormale orientiert. Der Benutzer hat die Wahl, die Objekte so snappen zu lassen, daß die Flächennormalen in dieselbe oder in die entgegengesetzte Richtung zeigen.

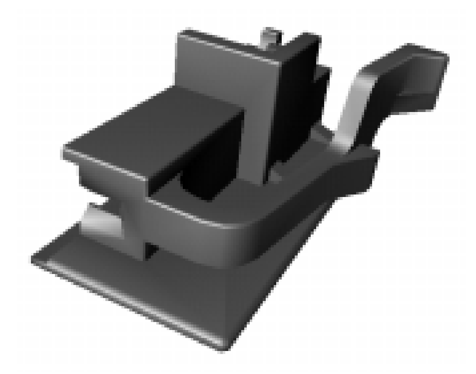
4.2.10.6 Performanzvergleich

Um die Leistungsfähigkeit des beschriebenen Verfahrens zu evaluieren, wurden Tests durchgeführt und die Ergebnisse mit den Ray-Pick-Algorithmen, die in OpenInventor [178] bzw. ACIS [145] realisiert sind, verglichen. Die Berechnung des nächstgelegenen Punktes innerhalb des vorgestellten Verfahrens erfolgte unter Zuhilfenahme der *findNearestPoint*-Methode von ACIS. Der Benchmark wurde mit drei Modellen bzw. Szenen unterschiedlicher Komplexität durchgeführt, die die Charakteristika der Verfahren aufzeigen.

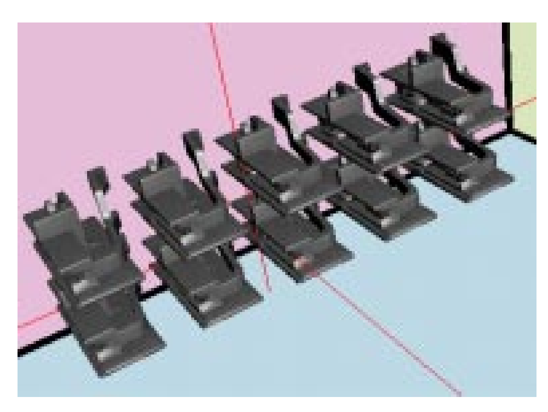
Das Modell A bildete ein prismatischer Kegelstumpf, der aus 50 Flächen bestand; 48 Flächen für die Mantelfläche plus zwei Flächen für Boden und Deckel (siehe Abbildung 68a). Als Modell B kam ein Kunststoffspritzgußteil zum Einsatz, das 159 topologische Flächen besaß. Aus der Facettierung des Modells gingen 989 Facetten hervor, die die graphische Repräsentation des Modells in OpenInventor bildeten (siehe Abbildung 68b). Die dritte Szene (siehe Abbildung 68c) wurde aus zehnmal Modell B gebildet und läßt Aussagen über das Verhalten des Algorithmus in Zusammenbaumodellen (assemblies) zu.



a) Modell A: ein prismatischer Kegelstumpf mit 50 Flächen



b) Modell B: ein Spritzgußteil mit 159 topologischen Flächen bzw. 989 Facetten



c) Szene bestehend aus 10mal Modell B

Abbildung 68: Testmodelle für den Picking-/Snapping-Algorithmus

Der Benchmark wurde auf einer SGI Indigo2 mit einem R4400 mit 150 Mhz ausgeführt. Der Rechner hatte 128 MB RAM und eine Extreme Graphikkarte. Die Applikationsgeschwindigkeit der Graphik unter OpenInventor beträgt maximal 200 bis 250 Tausend Dreiecke pro Sekunde.

Pick-Performanz

Die Pick-Performanz des vorgestellten 3D-Pick-Verfahrens wurde mit dem ray pick von ACIS und OpenInventor verglichen. Zunächst erfolgte ein wiederholtes Picken auf Modell B, um festzustellen, ob die Laufzeit mit der Anzahl der Picks variiert. Dabei wurde jeweils dreimal eine Fläche, Kante und Ecke selektiert. Der Test ergab, daß die Laufzeitcharakteristik der drei Algorithmen vom Typ des gepickten Elementes unabhängig ist, so daß auf eine Darstellung der Ergebnisse für Ecken und Kanten verzichtet werden kann.

Tabelle 5 zeigt die Ergebnisse der drei Verfahren für den Flächenpick. Zu erkennen ist, daß OpenInventor über die drei Versuche ein quasi konstantes Laufzeitverhalten zeigt. ACIS und der 3D-Pick zeigen ab dem zweiten Versuch ein derartiges Verhalten (Tests mit mehr als dreimaliger Wiederholung bestätigten dies). Die vergleichsweise langsame Ausführung des ersten Picks liegt im Aufbau interner Datenstrukturen begründet. Der 3D-Pick ist beim ersten Mal am langsamsten, übertrifft ACIS ab dem zweiten Pick aber um den Faktor 2-3. Die Verzögerung beim ersten Pick ist auf den Datenstrukturaufbau und die Berechnung der BoundingBox zurückzuführen, die in der aktuellen Implementierung des 3D-Pick-Verfahrens on-demand erfolgen. Beides ließe sich für alle

Elemente der Szene schon im voraus berechnen und ablegen, wodurch sich dieser Effekt auf Kosten des Speicherplatzbedarfs eliminieren ließe.

Tabelle 5: Pick-Performanz beim Flächenpick auf Modell B abhängig von der Anzahl der Picks

	1. Pick	2. Pick	3. Pick
OpenInventor ray pick	0,04700 sec.	0,05983 sec.	0,04851 sec.
ACIS ray pick	0,07214 sec.	0,02286 sec.	0,02279 sec.
3D-Pick	0,09042 sec.	0,01034 sec.	0,00707 sec.

Abbildung 69 stellt obige Tabelle graphisch dar.

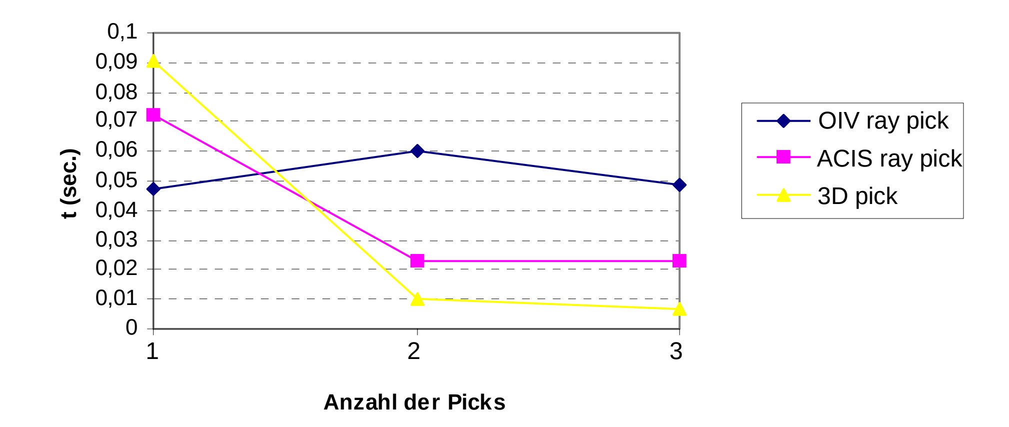


Abbildung 69: Laufzeit der Picking-Algorithmen in Abhängigkeit von der Anzahl der Picks

Der nächste Test gilt dem Laufzeitverhalten in Abhängigkeit von der Szenenkomplexität. Tabelle 6 zeigt die benötigte Berechnungszeit der drei Verfahren für die unterschiedlichen Szenen. Bei Verfahren zwei und drei wurden die Zeiten, die für den zweiten und dritten Pick benötigt wurden, gemittelt, da das Zeitverhalten beim ersten Pick untypisch ist (s.o.).

Tabelle 6: Laufzeit des Pick-Verfahrens in Sekunden abhängig von der Szenenkomplexität

	Modell A, 50 Facetten	Modell B, 1.000 Facetten	10mal Modell B, ca. 10.000 Facetten
OpenInventor ray pick	0,004180	0,051780	0,557720
ACIS ray pick	0,012550	0,019265	0,022825
3D-Pick	0,006135	0,006750	0,008650

Aus der Tabelle und Abbildung 70 geht hervor, daß OpenInventor ein mit der Szenenkomplexität lineares Zeitverhalten aufweist. Der ACIS ray pick sowie die vorgestellte 3D-Pick-Methode zeigen ein annähernd konstantes Verhalten, wobei der 3D-Pick im Durchschnitt 2,5mal schneller ist als der ACIS ray pick. Zu beachten ist, daß OpenInventor im ersten Fall - bei dem Modell mit nur 50 Faceten - einen Geschwindigkeitsvorteil besitzt.

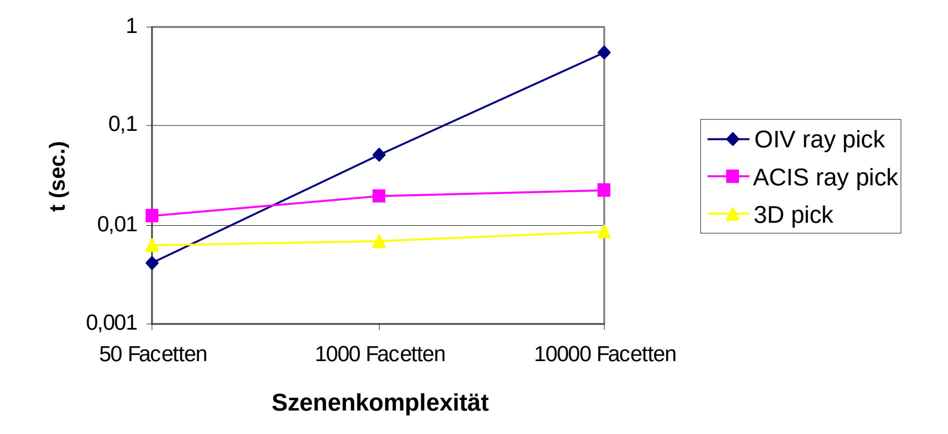


Abbildung 70: Laufzeit der Picking-Algorithmen in Abhängigkeit von der Szenenkomplexität

Auch die durchschnittliche Dauer der einzelnen Berechnungsschritte des Algorithmus wurde gemessen. Dabei stellte sich heraus, daß jeder Schritt des Algorithmus zur Vermeidung des nächst komplexeren Berechnungsschrittes etwa eine Größenordnung weniger rechenzeitintensiv, d.h. ein BoundingBox-Check kann ca. 10mal so schnell ausgeführt werden wie die Prüfung, ob eine Neuberechnung des nächsten Punktes nötig ist, und die kann wiederum ca. 10mal so schnell ausgeführt werden, wie die Neuberechnung selbst. Da die Dauer der Berechnung eines nächsten Punktes mit der Komplexität der Geometrie des topologischen Elementes variiert, zu dem der Punkt gehört, können hier Schwankungen auftreten.

Performanz des Snapping-Algorithmus

Um das Laufzeitverhalten des Snappings zu untersuchen, wird folgender Versuch unternommen: Der 3D-Cursor wird entlang der Hauptdiagonalen der BoundingBox um das Modell B programmatisch in 100 äquidistanten Schritten bewegt. Durch jeden Bewegungsschritt wird der Snapping-Algorithmus angestoßen; dessen Berechnungszeit wird über die hundert Schritte kumuliert. Um zu praxisgerechten Aussagen zu gelangen, ist das Snapping für Flächen, Kanten und Ecken aktiviert.

Auf den drei Raumdiagonalen werden in 27 bis 70 Fällen nächstgelegene Punkte innerhalb des Pickradius gefunden. Die Berechnungszeit des Snapping-Algorithmus über die 100 Schritte beträgt über alle drei Diagonalen gemittelt 1,0297 Sekunden, so daß der Snapping-Algorithmus theoretisch eine Bildwiederholrate von knapp 100 Hz zulassen würde - bei einem Modell mit ca. 1000 Facetten. Für ein interaktives Verhalten sind jedoch 20 Hz ausreichend.

Neben den objektiven Rechenzeiten ist auch der subjektive Eindruck entscheidend. Konnte mit dem naiven Ansatz, bei dem mit jeder Cursorbewegung die nächstgelegenen Punkte aller Objekte und ihrer topologischen Elemente bestimmt wurden, und anschließender Selektion des Punktes mit dem minimalen Abstand zum 3D-Cursor, nur in sehr kleinen, praxisfremden Szenen gearbeitet werden [99], so ist durch das beschriebene Verfahren zum schnellen Picken und Snappen mit einem 3D-Cursor nun auch das Arbeiten mit praxisnahen CAD-Modellen, wie dem obigen Spritzgußteil, möglich. Dabei gleitet der Cursor im Snapping-Modus quasi verzögerungsfrei über die Objektoberfläche und ermöglicht somit das präzise Konstruieren basierend auf vorhandenen Objekten. Das vorgestellte Verfahren hat einen wesentlichen Einfluß auf die mit 3D-Eingabegeräten erzielbare Arbeitsgeschwindigkeit in Modellierungsaufgaben (siehe Kapitel 5.1).

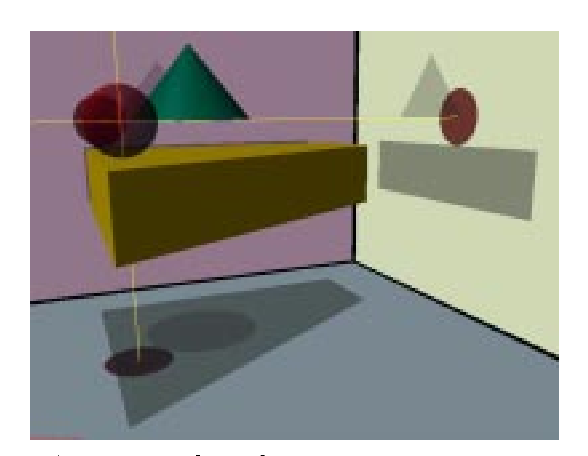
Das Snapping ist prinzipiell auch dazu geeignet, ein Kraftrückkopplungsgerät zu treiben. Dabei entfiele durch direkte Übergabe der Snap-Position an das Kraftrückkopplungsgerät sogar die heute oft noch umständliche Ansteuerung über einen speziellen haptischen Szenengraphen, der mit der Szene konsistent gehalten werden muß.

Mit dem vorgestellten Verfahren werden bekannte Pick- und Snap-Funktionalitäten auf 3D erweitert und in einer Weise beschleunigt, daß sie direkt in 3D durchgeführt werden können, ohne den Interaktionsprozeß zu bremsen und somit eine Hilfestellung beim präzisen Modellieren bieten. Auch in diesem Zusammenhang eröffnet der Einsatz von 3D-Eingabe neue Möglichkeiten. So kann über das Snapping hinaus eine neue Interaktionsform realisiert werden, nämlich *Repulsion*.

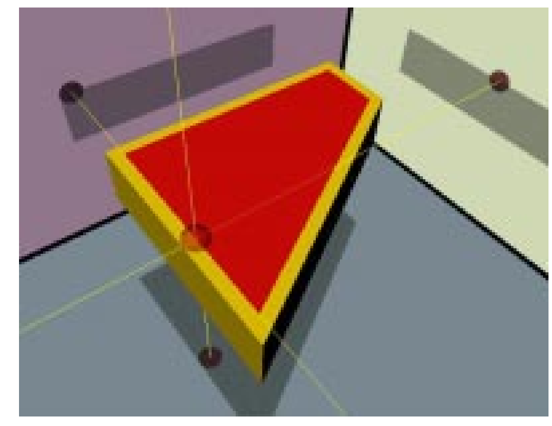
4.2.10.7 Repulsion

Repulsion - die Abstoßung des 3D-Cursors von der Objektoberfläche - erlaubt es, Objekte basierend auf vorhandenen Objekten in einem durch die Pickkugel definierten Abstand zu erzeugen und gleichzeitig die Orientierung von dem Basisobjekt zu übernehmen.

Repulsion basiert auf dem vorgestellten Snapping-Algorithmus. Anstatt den 3D-Cursor an die Position des nächstgelegenen Punktes zu setzen, wird die Flächennormale an diesem Punkt dazu benutzt, um den Cursor in Richtung der Flächennormale um den Betrag des Pickradius' 'wegzuschieben'. Dadurch scheint die Pickkugel des 3D-Cursors auf der Oberfläche des Modells zu rollen (siehe Abbildung 71a). Ist Repulsion während der interaktiven Objekterzeugung aktiv, so kann durch Berücksichtigung der Flächennormale das in der Erzeugung befindliche Objekt gemäß dem Basisobjekt orientiert werden, ohne es zu berühren (vgl. Kapitel 4.2.3).



a) Face-Repulsion bei Erzeugung eines 3D-Primitivs



b) Kombination von Face-Snapping und Edge-Repulsion zur Erzeugung einer Fläche mit Offset zu Kanten

Abbildung 71: Beispiele für die Anwendung von *Repulsion*

Durch Kombination von Snapping und Repulsion für verschiedene topologische Elemente ist es z.B. auch möglich, ein Polygon auf einer Fläche mit einem definierten Abstand von den Kanten der Fläche zu erzeugen (siehe Abbildung 71b).

4.2.11 Ein Attribut-basierter Ansatz zum systemunterstützten interaktiven Zusammenbau virtueller Modelle

Die bislang vorgestellten Interaktionstechniken unterstützen in erster Linie die schnelle und präzise Erstellung und Modifikation von Bauteilen mit einem 3D-Eingabegerät. Darüber hinaus ist der Umgang mit Zusammenbaumodellen (assemblies), also Teilmodellen, die zu einem Ganzen zusammengefügt werden, ein immer wichtiger werdender Bestandteil von CAD-Systemen.

Im Produktentwicklungsprozeß ist es wichtig zu prüfen, ob alle Teilmodelle zusammenpassen und ob der Zusammenbau des virtuellen Produktes aus den Teilmodellen möglich ist. Aufgrund der Bedeutung dieses Schrittes hat sich das Design Review zu einem Hauptanwendungsfall von VR-Systemen entwickelt. Im Design Review eines Digital Mock-Ups (digitalen Prototypen) kann geprüft werden, ob Bauräume eingehalten werden oder Teile kollidieren. Auch die Montage- bzw. Demontagesimulation ist ein wichtiges Einsatzgebiet für VR-Systeme [69].

VR-Systeme setzen zu diesem Zweck Verfahren zur Kollisionserkennung ein, d.h. das zu montierende Bauteil wird mit jeder Bewegung auf Kollisionen mit seiner Umgebung geprüft. Kann ein Teilmodell kollisionsfrei ein- oder ausgebaut werden, so kann geschlossen werden, daß mit hoher Wahrscheinlichkeit auch das reale Teil ein- bzw. ausbaubar ist. Neben dem Bauteil ist auch die Zugänglichkeit des Bauraumes für den Monteur zu beachten.

Die Zusammenbausimulation mittels Kollisionserkennung ist mit folgenden Problemen behaftet:

- Kollisionen können in Echtzeit nur auf triangulierten - und damit ungenauen - Modellen berechnet werden. Aus diesem Grund kann es vorkommen, daß zwei triangulierte Modelle nicht kollidieren, obwohl die exakten CAD-Modelle, aus denen sie hervorgegangen sind, kollidieren würden (siehe Abbildung 72).
- Mit der Kollisionsberechnung kann nur erkannt werden, wann zwei Teile kollidieren, d.h. die Kollisionsfreiheit zweier Teile sagt nichts über die Paßgenauigkeit und das Erreichen der Einbaulage aus.
- Die Kollisionserkennung gibt dem Benutzer keine aktive Unterstützung während der Einbausimulation, d.h. der Benutzer muß das Teil solange bewegen, bis er meint, die Einbaulage erreicht zu haben. Da dies allein auf dem Wege der visuellen Rückkopplung kaum möglich ist, behilft man sich in VR-Systemen oft damit, die Einbaulage dem zu montierenden Teil mitzugeben. So kann während der Einbausimulation geprüft werden, ob diese annähernd erreicht ist und falls ja, das Teil automatisch in die gewünschte Endlage gebracht werden [69]; problematisch wird dieser Ansatz, wenn die Einbaulage unbekannt ist.

Abbildung 72 veranschaulicht das Genauigkeitsproblem der Kollisionserkennung mit triangulierten Objekten. Links in der Abbildung ist die mathematisch exakte Ausgangssituation dargestellt. Ein Bolzen ist paßgenau in ein Volumen mit entsprechender Bohrung eingeführt. Bolzen und Bohrung haben denselben Radius; in der Praxis ist selbstverständlich eine geringe Toleranz nötig. Die rechte Abbildung zeigt den Zylinder und das umgebende Material in ihrer facettierten Form. Da die Facettierung sich der Oberfläche stets von innen (der Seite des Materials) nähert, entsteht ein Spalt. Dies bedeutet, daß auch ein geringfügig größerer Zylinder in seiner facettierten Form in das facettierte Loch passen würde, nicht jedoch in seiner exakten Form in die exakte Bohrung.

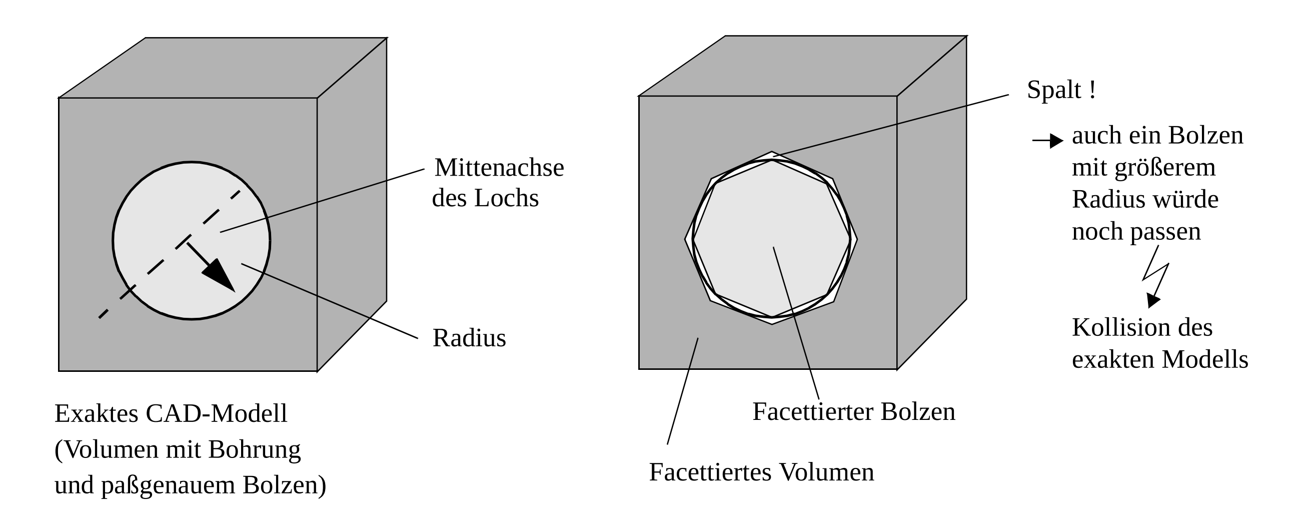


Abbildung 72: Auswirkung der Facettierungsgenauigkeit auf die Kollisionserkennung

Für den geschilderten Fall sind Dreiecks-basierte Kollisionserkennungsverfahren viel rechenzeitintensiver als Ansätze, die sich der Attribute (oder auch Features) des mathematisch korrekten CAD-Modells bedienen und das Bolzen-Bohrungs-Problem durch einfachen Vergleich der Position, Orientierung und des Radius' lösen könnten. Solche Lösungsansätze sind aus der Literatur als *design for assembly/disassembly* bekannt [124].

Design for assembly/disassembly basiert i.d.R. auf Assembly-Features [138],[139]; Assembly-Features beschreiben zwischen Teilmodellen vorkommende Verbindungstypen. Zunächst müssen alle Verbindungstypen, die verwendet werden sollen, erfaßt und als Assembly-Feature modelliert werden. Der Konstruktionsprozeß muß unter Verwendung der Assembly-Features erfolgen. Die Assembly-Features (siehe Abbildung 73) können anschließend dazu verwendet werden, um Zusammenbauprozesse zu simulieren oder die Freigängigkeit von Bauteilen zu untersuchen [5].

Assembly-Features besitzen neben einer geometrischen Ausprägung eine Reihe von zusätzlichen semantischen Informationen, die dazu herangezogen werden, die genannten Funktionalitäten zu realisieren. Von besonderer Bedeutung ist das lokale Koordinatensystem eines Assembly-Features, dessen Position so gewählt ist, daß es die gewünschte Verbindungsfunktionalität bestmöglich unterstützt. Der Einsatz von Assembly-Features bedarf also gewissen Vorplanungen und der Einhaltung bestimmter Prinzipien in der Konstruktionsphase.

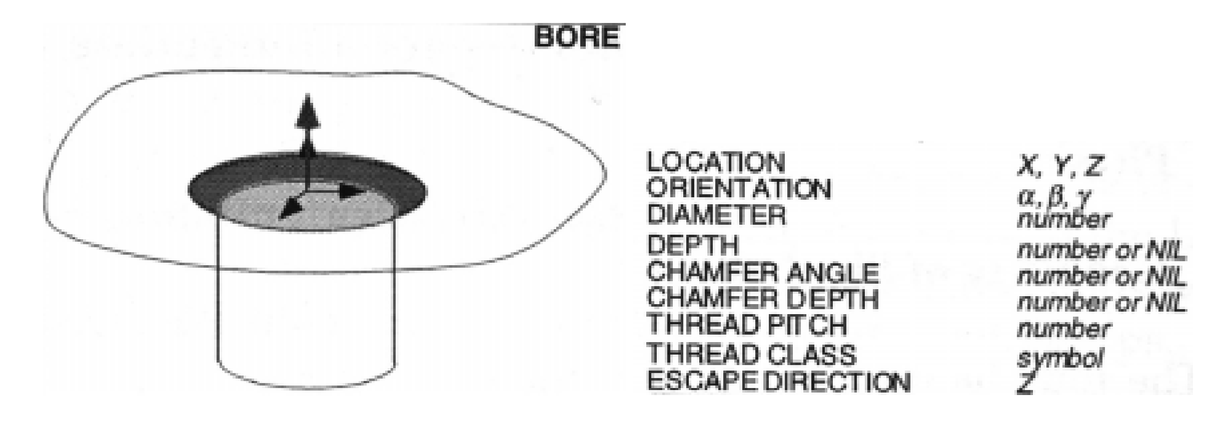


Abbildung 73: Beispiel eines Assembly-Features und seiner Semantik [180]

Whitney [180] sagt in seiner Arbeit über zukünftige CAD-Werkzeuge im Zusammenhang mit der Bauteilmodellierung:

> *„... The tools and information gathering activities should not encumber the designers or cause them additional effort. If this happens, the designers will turn them off. ...“*

Aus diesem Grund wird nachfolgend ein Attribut-basierter Ansatz zur Unterstützung des interaktiven Zusammenbauprozesses (Assembling Assistance) konzipiert, der den Benutzer mit weniger Vor- und Aufbereitungsaufwand in der Modellierungsphase konfrontiert und somit einen Beitrag zur Benutzerzentriertheit eines Modelliersystems auch für die Zusammenbaumodellierung leistet [159].

Die grundlegende Idee des Ansatzes ist, die Geometrie während des Konstruktionsprozesses automatisch mit Attributen anzureichern und diese Attribute später heranzuziehen, um dem Benutzer eine Unterstützung beim Zusammenbau zu bieten. Die Unterstützung besteht im automatischen Ausrichten und Limitieren der Bewegungsfreiheitsgrade eines Bauteils während des interaktiven Zusammenbauprozesses. Auch das Erreichen die Einbaulage kann auf Basis der Attribute erkannt werden.

Bei dem Verfahren werden zwei Phasen unterschieden, nämlich die Anreicherung und Behandlung von Primitiven mit Attributen während der Bauteilmodellierung sowie die eigentliche Assistenzphase in der interaktiven Zusammenbausimulation. Während der Bauteilmodellierung werden von dem Verfahren folgende Schritte vorgenommen:

1. Attributierung der Primitive und Zuweisung der Attribute an topologische Elemente
2. Behandlung der Attribute während Boolscher Operationen

Als Attribute kommen Flächen, Kanten, Achsen und Parameter zum Einsatz.

Soll nun ein attributiertes Modell interaktiv unter Benutzung eines 3D-Eingabegerätes zusammengebaut werden, ist mit jeder Bewegung zu prüfen, ob sich das bewegte Modell in der Nähe eines korrelierenden Modells befindet. Falls ja, wird es entsprechend orientiert und die Freiheitsgrade werden eingeschränkt. Diese zweite Phase läuft in folgenden Schritten ab:

1. Prüfen der Umgebung des bewegten Teilmodells auf korrelierende Attribute
2. Orientieren und Positionieren des Teilmodells
3. Prüfen auf Durchdringung

Das Verfahren kombiniert - in gewisser Weise - die Idee von Assembly-Features mit der Idee der allowable-motion [55], wobei Assembly-Features nicht explizit verwendet werden müssen, sondern implizit erzeugt werden. Gegenüber der allowable-motion-Methode besteht der Vorteil, daß Bewegungen auch nach einer Seite eingeschränkt werden können[5], z.B. entlang der Mittelachse eines Sackloches, und die Einbaulage ohne Kollisionserkennung im herkömmlichen Sinne auf den mathematisch exakten Attributen erkannt werden kann. Weiterhin besteht die Möglichkeit, mit Formtoleranzen umzugehen, in dem die Parameter mit Toleranzen versehen werden.

5. Whitney [180] wies in seiner Arbeit darauf hin, daß für die Modellierung solcher einseitiger Bewegungseinschränkungen noch Handlungsbedarf für weitere Forschungsarbeiten besteht.

Diesen Vorteilen stehen folgende Einschränkungen bzw. Nachteile gegenüber:

- Das Verfahren basiert auf einem Grundkörper-orientierten Vorgehen.
- Selbst bei Beschränkung auf Primitive mit planaren und zylindrischen Flächen[6], kann es rasch zu Situationen kommen, bei denen der Versuch, Durchdringungen festzustellen, sehr berechnungsaufwendig oder gar unmöglich wird; in diesem Fall ist der Betrachter gefordert, der Durchdringungen visuell oft leicht erkennen kann.
- Das Verfahren arbeitet lokal, d.h. die beiden in einem Zusammenbauschritt involvierten Bauteile können an Stellen kollidieren, die nicht im Betrachtungsbereich liegen.

Dem Ansatz kommt die Tatsache entgegen, daß Verbindungen zwischen Bauteilen oft eine einfache Geometrie aufweisen, so daß er trotz seiner Einschränkungen in vielen Fällen eine Hilfestellung bieten kann. Eine Studie bei Mercedes Benz [128] zeigt, daß Bauteile im wesentlichen durch Verschraubungen, Verklebungen, Steck- und Schweißverbindungen zusammengehalten werden. Abbildung 74 zeigt beispielhaft verschiedene Steckverbindungen.

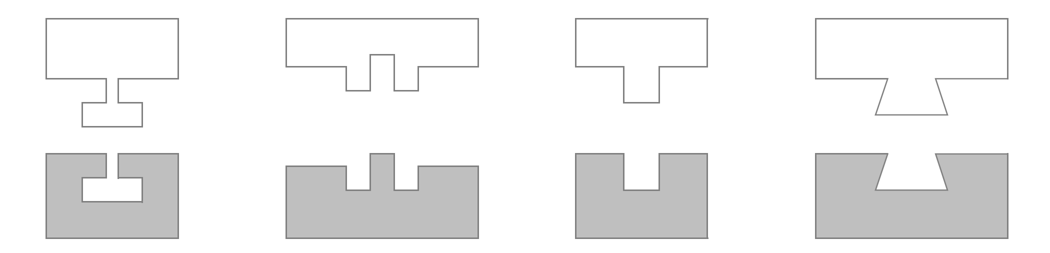


Abbildung 74: Beispielhafte Steckverbindungen [128]

Im Rahmen der vorliegenden Arbeit wurde das vorgestellte Konzept für zylindrische Steckverbindungen implementiert. Diese erste Implementierung zeigt die Machbarkeit eines solchen Ansatzes, wirft aber auch Fragen für zukünftige Forschungsaktivitäten auf. Um die einzelnen Schritte des Verfahrens zu verdeutlichen, werden diese im folgenden anhand eines Beispiels detailliert dargestellt; zunächst die Schritte, die während der Bauteilmodellierung ablaufen.

4.2.11.1 Automatische Attributierung in der Modellierungsphase

Bei der Erzeugung von Primitiven müssen nicht nur die entsprechenden Attribute generiert werden, sondern auch dem richtigen topologischen Element des Primitivs zugeordnet und unter Boolschen Operationen beachtet werden.

Für einen Zylinder bedeutet dies, daß die Attribute ’Hauptachse’ und ’Radius’ erzeugt werden müssen. Der Radius r hat einen Gültigkeitsbereich von

$$p_0 + t \cdot (p_1 - p_0)\ , \text{ mit } t \in [0, 1]\ ,$$

wobei p_0 und p_1 gleich Mittelpunkt der Boden- bzw. Deckelfläche (siehe Abbildung 75).

Die Attribute ’Hauptachse’ und ’Radius’ werden der Mantelfläche zugeordnet, da sie im Gegensatz zu Boden- und Deckelfläche unter subtraktiv wirkenden Boolschen Operationen

6. Fa [55] beschränkt sich ebenfalls auf planare und zylindrische Flächen und führt keine Durchdringungsbetrachtung durch.

erhalten bleibt. Aus der Attributkombination geht auch nach Boolschen Operationen noch hervor, daß es sich um einen Zylinder handelt (siehe Abbildung 75).

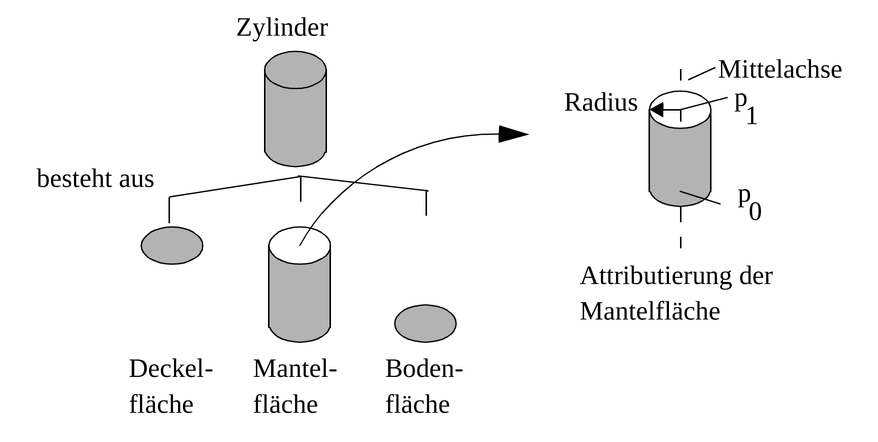


Abbildung 75: Attributierung eines Zylinders

4.2.11.2 Attributverarbeitung in der Modellierungsphase

Werden attributierte Primitive Boolschen Operationen unterzogen, so sind zwei Bearbeitungsschritte notwendig:

1. Beachtung und Behandlung verschiedener Modellkoordinatensysteme.
2. Auflösung möglicher Konflikte unter den Attributen.

Für eine Bohrung bedeutet dies, daß das Attribut des die Bohrung definierenden Zylinders an das Ergebnis der Boolschen Operation ’lagerichtig angehängt’ werden muß; das Attribut wird mit dem topologischen Element an das Ergebnis ’vererbt’.

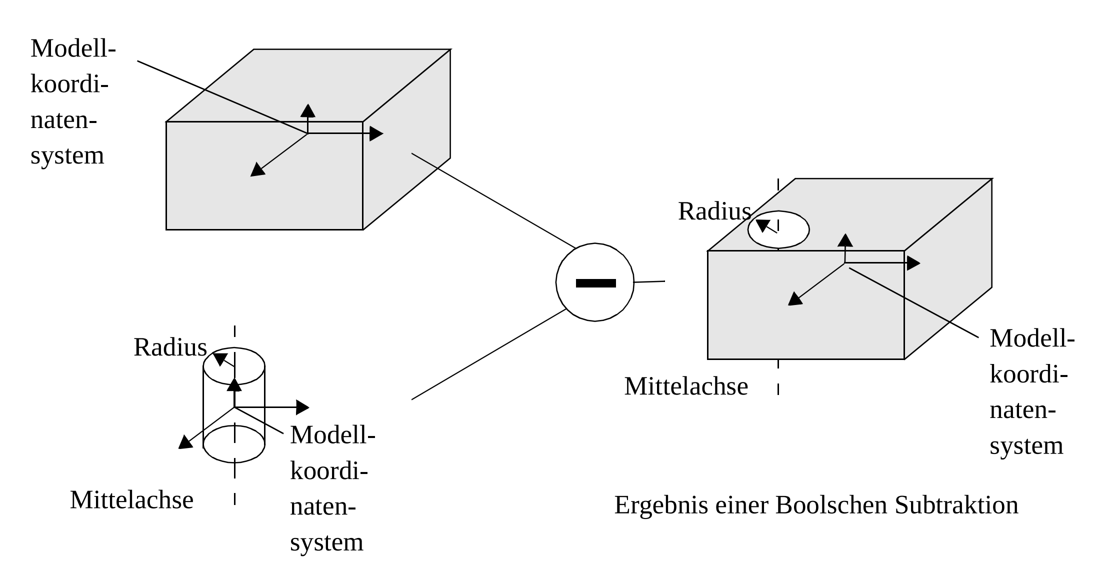


Abbildung 76: Beachtung verschiedener MKS bei der Attributvererbung während Boolscher Operationen

Abbildung 76 verdeutlicht diesen Verfahrensschritt: Links sind die beiden Primitive Zylinder und Quader zu sehen. Der Ursprung der Modellkoordinatensysteme (MKS) beider Primitive liegt jeweils in deren Zentrum. Das MKS des Ergebniskörpers (rechts) resultiert aus dem MKS des Quaders, da von ihm der Zylinder subtrahiert wird, um eine Bohrung zu erzeugen. Die Bohrung

im Ergebniskörper liegt exzentrisch zum MKS, so daß die Attribute des Zylinders bzw. seiner Mantelfläche entsprechend transformiert werden müssen, um auch im Ergebnis ihre Gültigkeit zu behalten.

Neben der Transformation von Attributen kann auch ein ’Verschmelzen’ von Attributen während Boolscher Operationen notwendig werden. Dies ist beispielsweise der Fall, wenn aus zwei Zylindern mit unterschiedlichen Radien ein Bolzen erzeugt werden soll. Abbildung 77 zeigt, wie die kollinearen Attribute ’Hauptachsen’ in einem solchen Fall behandelt werden. Die beiden Hauptachsen werden zu einer zusammengefaßt und die von den Ausgangsprimitiven vererbten Radien erhalten entsprechende Gültigkeitsbereiche.

Der Radius r_0 hat einen Gültigkeitsbereich von $p_0 + t \cdot (p_i - p_0)$, mit $t \in [0, 1]$.

Der Radius r_1 hat einen Gültigkeitsbereich von $p_i + t \cdot (p_1 - p_i)$, mit $t \in]\, 0, 1\,]$.

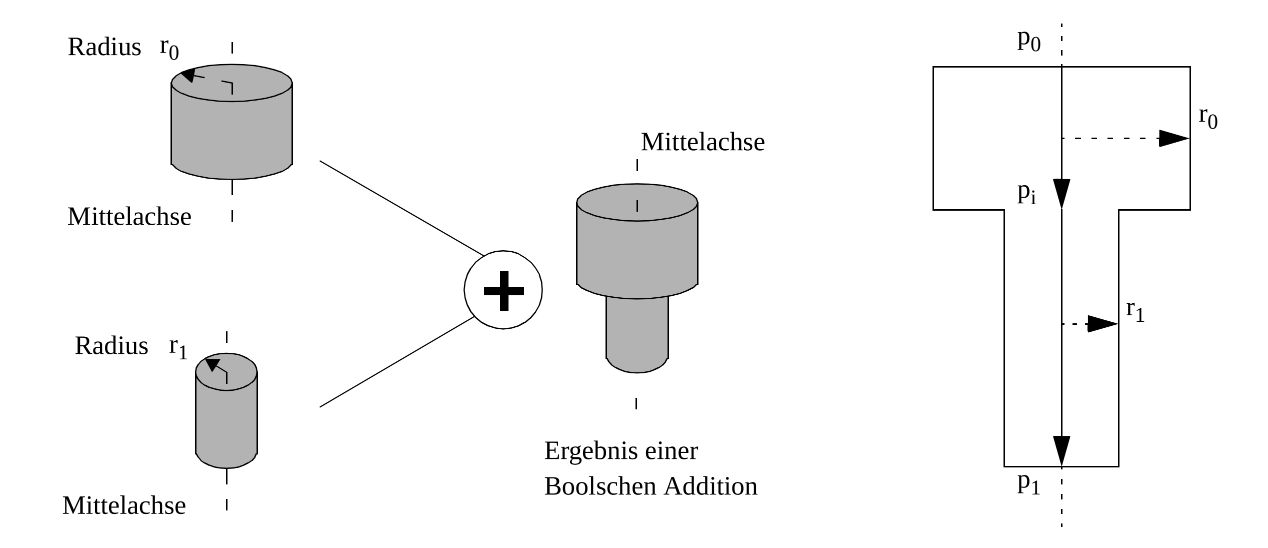


Abbildung 77: Kollineare Achsen und Gültigkeitsbereiche verschiedener Radien

Bei kollinearen Achsen können in einem Gültigkeitsbereich zwei oder mehr verschiedene Radien auftreten (siehe Abbildung 78b). Dieser Fall kann nach einer Boolschen Subtraktion, z.B. bei der Erstellung eines Lagers auftreten (siehe Abbildung 78a). Um ’Löcher’ von ’Material’ zu unterscheiden, kann der aus der Boolschen Subtraktion hervorgegangene ’innere’ Radius des Ergebniskörpers als negativ begriffen werden; auch mehr als zwei Radien in einem Gültigkeitsbereich sind möglich (siehe Abbildung 78c).

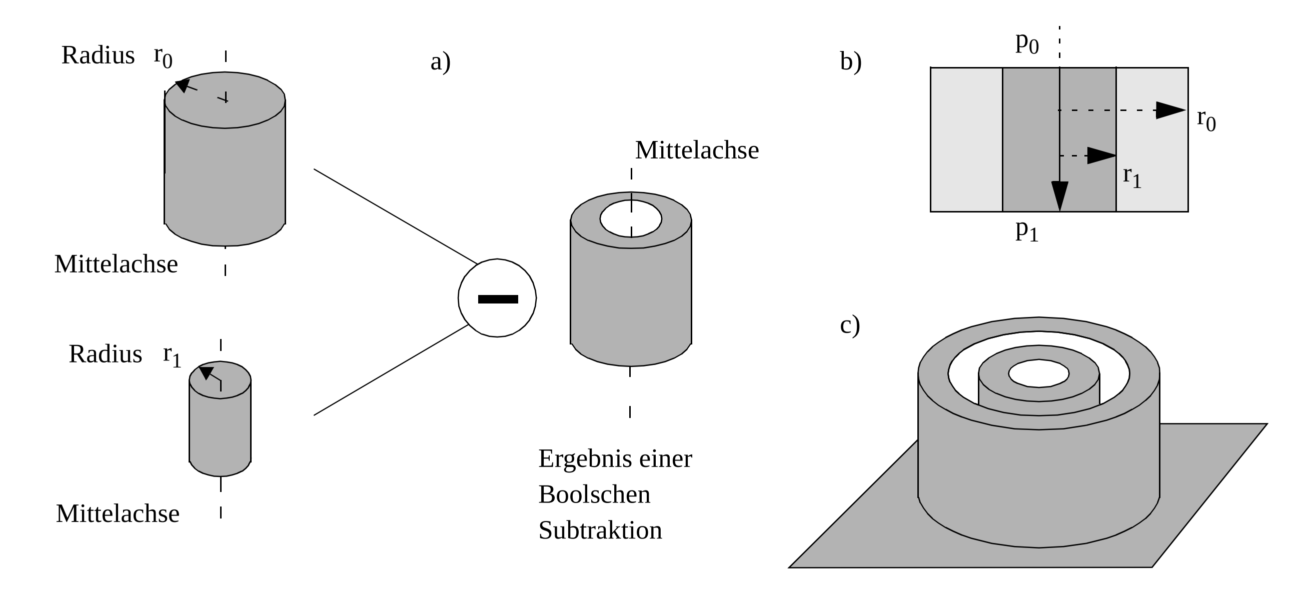


Abbildung 78: Verschiedene Radien in einem Gültigkeitsbereich

Die bislang beschriebenen Schritte betreffen die während der Bauteilmodellierung durchgeführten Maßnahmen, um den interaktiven Zusammenbau von Teilen zu unterstützen.

4.2.11.3 System-unterstützte Zusammenbausimulation

Im folgenden wird gezeigt, wie die zusätzliche Attributinformation in der interaktiven Zusammenbausimulation mit einem 3D-Eingabegerät genutzt wird, um einen Bolzen in ein Lager einzuführen und in Endlage zu bringen. Ohne jegliche Systemunterstützung ist es nahezu unmöglich, ein Bauteil graphisch-interaktiv exakt in seine Einbaulage zu bringen (siehe Abbildung 79). Für den Fall, daß das Eingabegerät nur 3 Freiheitsgrade besitzt, kann die Systemunterstützung die fehlenden drei Freiheitsgrade durch automatische Orientierung des Bauteils kompensieren.

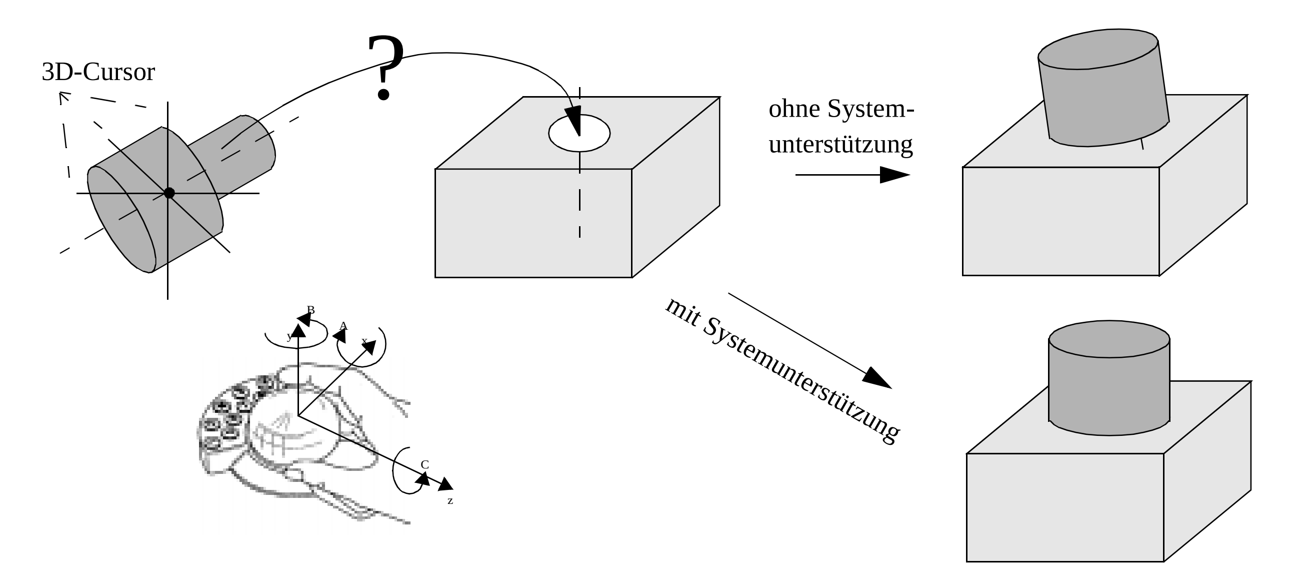


Abbildung 79: Notwendigkeit der Systemunterstützung bei der Zusammenbausimulation

Die interaktive Zusammenbausimulation läuft in folgenden Schritten ab:

1. Selektieren und Bewegen des zu montierenden Teils
2. Finden korrelierender Attribute mit Hilfe des 3D-Picks
3. Orientieren und Positionieren des Teilmodells
4. Einschränken der Bewegungsfreiheitsgrade
5. Kollisionsprüfung auf Basis der Attribute

Selektieren und Bewegen des zu montierenden Teils

Das zu montierende Teil wird mit dem 3D-Eingabegerät selektiert. Dazu wird der 3D-Cursor so plaziert, daß er das Attribut schneidet, welches im weiteren Verlauf des Zusammenbauprozesses betrachtet werden soll[7]. Bezogen auf das Beispiel heißt das, daß der Bolzen an seiner Mittelachse gepickt wird. Der Mittelachse sind zwei Radien zugeordnet. Ist das Objekt selektiert, kann es mit 3 bzw. 6 Freiheitsgraden - je nach 3D-Eingabegerät - im Raum bewegt werden.

Finden korrelierender Attribute mit Hilfe des 3D-Picks

Während das Teil bewegt wird, wird ähnlich wie beim Objekt-Objekt-Snapping (vgl. Kapitel 4.2.10) mit jeder Bewegung die Umgebung des gepickten Attributs auf korrelierende Attribute hin untersucht. Die Umgebung ist dabei über einen Pickradius definiert. Korrelierende Attribute können im allgemeinen über ihre Werte erkannt werden; auch die automatische Zuweisung eines Attributtyps bei der Erzeugung eines Attributs während der Modellierung ist möglich, wurde aber nicht realisiert, da die Implementierung - wie bereits erwähnt - bislang nur zylindrische Verbindungen unterstützt.

Für das Beispiel bedeutet dies, daß die Mittelachse der Bohrung bei Annäherung des Bolzens durch den kontinuierlichen Attribut-Pick gefunden wird. Daß es sich bei dem der Bohrung zugeordneten Attribut, um ein korrelierendes Attribut handelt, kann anhand des negativen Wertes für den Radius festgestellt werden (ein negativer Wert für einen Radius steht für eine zylindrische Bohrung).

Orientieren und Positionieren des Teilmodells

Hat das bewegte Teil ein korrelierendes Attribut erreicht, so wird es automatisch entsprechend orientiert und positioniert. Für die Orientierung wird der Rotationswinkel minimal gewählt. Für das Beispiel in Abbildung 80 bedeutet dies, daß der Bolzen um den Winkel α rotiert wird.

7. Zu diesem Zweck wurde der 3D-Pick so erweitert, daß auch Attribute selektiert werden können [113].

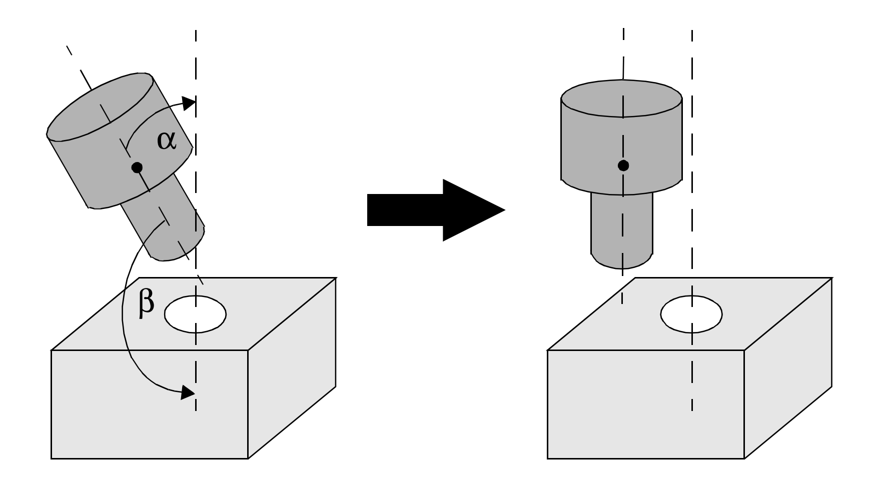


Abbildung 80: Automatisches Orientieren des Bolzens

Nachdem der Bolzen richtig orientiert ist, wird er auf die korrelierende Achse transliert. Dazu wird der Lotfußpunkt auf die durch die Mittelachse des Loches bestimmte Gerade berechnet und das Bauteil dorthin verschoben. Um die Berechnung zu erläutern und die nachfolgende Durchdringungsprüfung verständlich zu machen, werden zunächst einige Variablen eingeführt (vgl. Abbildung 81):

- Die Cursorposition *cp*.
- Der Richtungsvektor *r* definiert durch die Punkte p_0 sowie p_1, die das Intervall angeben, indem der Radius der Bohrung gültig ist.

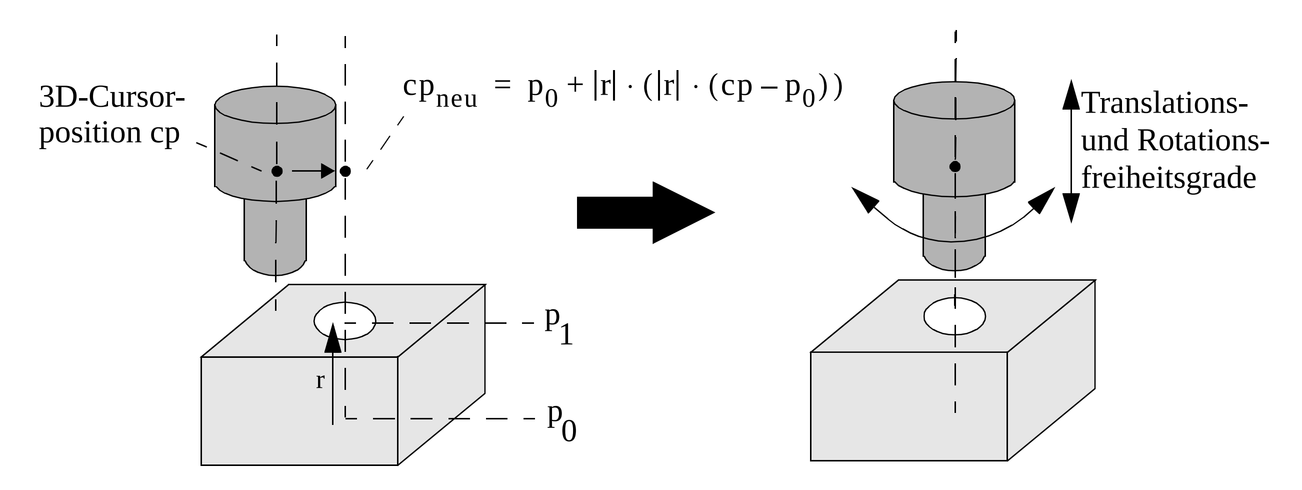


Abbildung 81: Automatisches Positionieren des Bolzens

<u>Einschränken der Bewegungsfreiheitsgrade</u>

Ist der Bolzen orientiert und positioniert, so werden die Translations- und Rotationsfreiheitsgrade eingeschränkt und es folgt mit jeder Bewegung eine Prüfung auf Durchdringung. Bei zylindrischen Steckverbindungen ist nur noch eine Translation entlang der Mittelachse bzw. eine Rotation um die Mittelachse möglich (siehe Abbildung 81). Zu beachten ist, daß diese Einschränkungen nur solange gelten, solange ein korrespondierendes Attribut im Pickraum gefunden wird. Bewegt

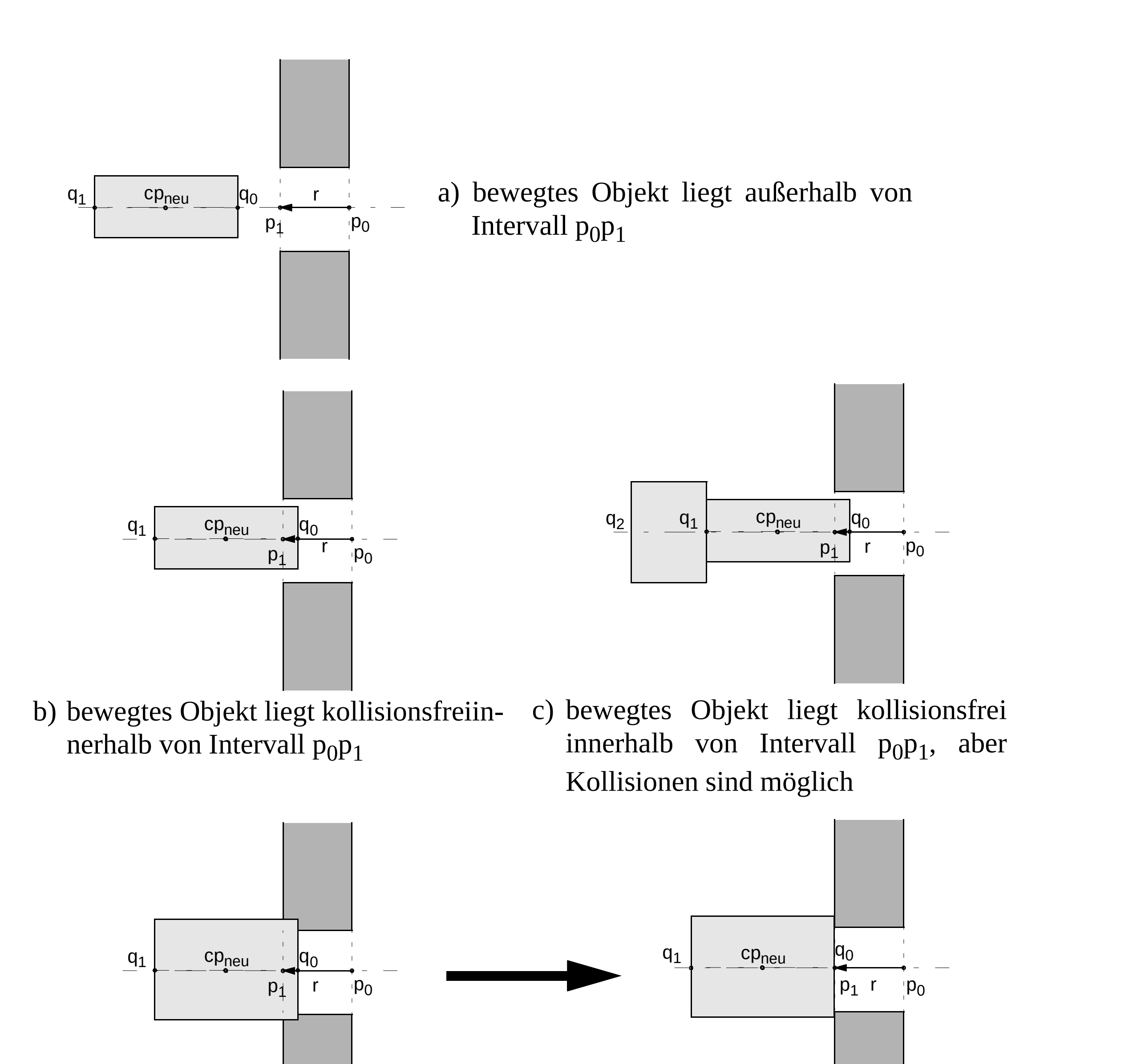
der Benutzer den Bolzen von dem Loch weg, so erlangt dieser bei hinreichender Entfernung seine vollen Freiheitsgrade zurück.

Prüfung auf Durchdringung

Die Prüfung auf Durchdringung zwischen dem Zylinder/Bolzen und dem Loch kann durch Vergleich der Radien im Parameterbereich des korrespondierenden Attributs durchgeführt werden. Dabei können drei verschiedene Situationen auftreten:

- Der Bolzen liegt außerhalb des Lochs (siehe Abbildung 82a).
- Er liegt kollisionsfrei innerhalb des Lochs (siehe Abbildung 82b).
- Er dringt in das das Loch umgebende Material ein (siehe Abbildung 82c).

Der erste Fall ist unkritisch, da keine Kollision vorliegt. Der zweite Fall ist ebenfalls unkritisch, es kann im Laufe weiterer Aktionen allerdings zu einer Kollision kommen, da der Bolzen ggfs. nur bis zu einem bestimmten Maß in die Bohrung eingebracht werden kann (siehe Abbildung 82c). Im dritten Fall liegt eine Kollision vor, die basierend auf den Attributinformationen erkannt wird. Diese Kollision wird automatisch aufgelöst, indem der Zylinder soweit entgegengesetzt zur Einbaurichtung transliert wird, bis er kollisionsfrei ist. Für das Beispiel in Abbildung 82e ergibt sich der Verschiebevektor v zu $v = p_1 - q_0$ unter der Annahme, daß die zur attribut-tragenden Fläche adjazenten Flächen orthogonal zur Mittelachse stehen. Besitzt eine zylindrische Steckverbindung mehrere Radien und Gültigkeitsbereiche, so werden diese in der Kollisionsauflösung ebenfalls berücksichtigt. Die in Abbildung 82d dargestellte Situation ließe sich schon während des Prüfens der Umgebung auf korrelierende Attribute vermeiden, würde man das korrelierende Attribut enger fassen und nur solche Partner betrachten, die tatsächlich und nicht nur potentiell passen. Die Durchdringungsprüfung und Kollisionsauflösung ist ein wichtiger und notwendiger Schritt des Verfahrens, der auch im Laufe der weiteren Montage in Abbildung 82c zum Tragen kommt.



a) bewegtes Objekt liegt außerhalb von Intervall p_0p_1

b) bewegtes Objekt liegt kollisionsfreiinnerhalb von Intervall p_0p_1

c) bewegtes Objekt liegt kollisionsfrei innerhalb von Intervall p_0p_1, aber Kollisionen sind möglich

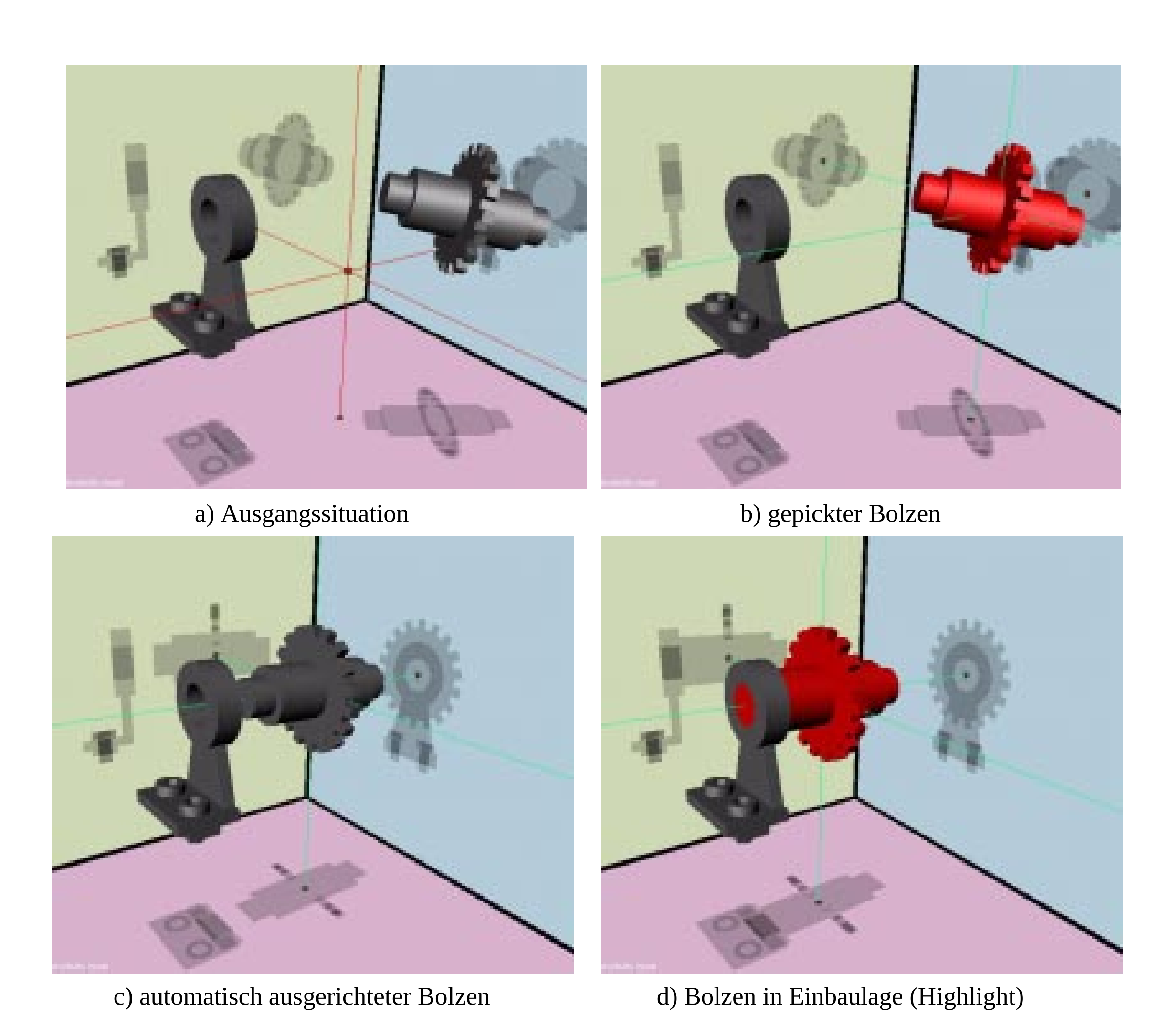
d) temporärer inkonsistenter Zwischenzustand nach Orientierungs- und Positionierungsphase

e) automatische Kollisionsaufhebung durch Verschieben des bewegten Objektes

Abbildung 82: Attribut-basierte Durchdringungsprüfung für zylindrische Verbindungen

Damit sind alle Schritte des Verfahrens beschrieben. Im folgenden wird das Verfahren für eine Montagesequenz eines Bolzens in einen Bolzenhalter nochmals zusammenhängend aus Benutzersicht dargestellt.

Abbildung 83a zeigt die Ausgangssituation mit einem Bolzenhalter und einem beliebig orientierten Bolzen. Der Bolzen besitzt unterschiedliche Radien, d.h. es existieren kollineare Hauptachsen mit verschiedenen Radien in unterschiedlichen Intervallen. Der 3D-Cursor ist in der Mitte der Szene zu sehen. Der Benutzer bewegt nun den 3D-Cursor zu dem Bolzen und selektiert diesen, woraufhin der Bolzen mit einem Highlight versehen wird (siehe Abbildung 83b).



a) Ausgangssituation
b) gepickter Bolzen
c) automatisch ausgerichteter Bolzen
d) Bolzen in Einbaulage (Highlight)

Abbildung 83: Assistierte Zusammenbausequenz

Anschließend beginnt der Benutzer den Bolzen in Richtung Bohrung des Bolzenhalters zu bewegen. Während jedes Bewegungsschrittes wird die Umgebung des Bolzens auf korrelierende Attribute hin untersucht. Nähert sich der Bolzen der Bohrung bis auf ein bestimmtes Maß, so wird diese anhand der Attributinformation als korrelierend erkannt. Der Bolzen wird automatisch gemäß der Mittelachse der Bohrung orientiert und positioniert; das Highlight verschwindet (siehe Abbildung 83c). Die Bewegungsmöglichkeit des Bolzens ist nun auf eine lineare Bewegung entlang der Hauptachse der Bohrung bzw. auf eine Rotation um die Hauptachse eingeschränkt. Der Benutzer bewegt den Bolzen weiter in Richtung Bohrung. Wenn sich Bolzen und Bolzenhalter berühren, wird das Highlight reaktiviert, um dem Benutzer anzuzeigen, daß eine mögliche und gültige Einbaulage erreicht ist (vgl. Abbildung 83d). Der Bolzen kann vom Benutzer nun entweder an dieser Stelle deselektiert oder wieder aus der Bohrung herausbewegt werden. Im letzten Fall erhält der Benutzer alle Bewegungsfreiheitsgrade erst dann zurück, wenn der Bolzen nicht mehr in dem 'Einzugsgebiet' der Mittelachse der Bohrung liegt.

4.2.11.4 Diskussion

Der vorgestellte Attribut-basierte Ansatz unterstützt den Benutzer während der Simulation interaktiver Zusammenbauprozesse. Die Unterstützung besteht in erster Linie in der automatischen Ausrichtung des zu montierenden Bauteils und einer entsprechenden Limitierung der dem

Benutzer verbleibenden Bewegungsfreiheitsgrade. Da der Ansatz mit exakten Attributen arbeitet, ist eine präzise Positionierung mit 3D-Eingabegeräten möglich.

Dies unterscheidet den Ansatz von der herkömmlichen Kollisionserkennung, die auf exakten CAD-Modellen nicht echtzeitfähig ist und nur eine unzureichende Benutzerunterstützung bietet, wenn ein Bauteil in Einbaulage gebracht werden soll. Von Assembly-Features unterscheidet sich der hier vorgestellte Ansatz durch die automatische Generierung und Propagierung von Attributen während der Modellierungsphase; es müssen nicht erst entsprechende Assembly-Features erstellt werden, sondern Volumenprimitiven werden automatisch Attribute zugeordnet, die im weiteren Modellierungsprozeß beachtet werden und schließlich der Benutzerunterstützung während der Zusammenbausimulation dienen.

Das Verfahren wurde im Rahmen der vorliegenden Arbeit für zylindrische Steckverbindungen implementiert. Im folgenden werden einige Ideen zu der Erweiterbarkeit des Ansatzes aber auch zu seinen Schranken diskutiert.

- Erweiterung auf andere Primitive

  Bei der Erweiterung des Verfahrens auf andere Primitive sind deren relevante Merkmale als Attribut auszudrücken und dem Primitiv zuzuordnen. Für einen Kegelstumpf ergäben sich beispielsweise eine Hauptachse und zwei Radien. Die Untersuchung auf korrelierende Attribute und die Prüfung auf Durchdringung muß entsprechend erweitert werden. Für den Kegelstumpf muß eine lineare Interpolation zwischen den beiden Radien während der Prüfung auf Durchdringung erfolgen. Die einzelnen Schritte des Verfahrens behalten bei einer Erweiterung auf andere Primitive ihre Gültigkeit.

- Integration von Toleranzen

  Da die Attribute im Sinne des CAD-Modells mathematisch exakt sind, lassen sich Toleranzen sowie die Behandlung von Toleranzen in den virtuellen Montageprozeß integrieren. Anstatt eines exakten Vergleichs von Attributinformationen kann ein Vergleich, der die Toleranzen miteinbezieht, stattfinden. Die Toleranzen müssen dem Modell vom Benutzer während bzw. nach der Modellierungsphase zugewiesen werden.

- Problem der Lokalität

  Das vorgestellte Verfahren arbeitet lokal. Ob zwei Modelle zusammenpassen, wird anhand eines Attributpaares entschieden. Auch wenn die Modelle innerhalb des lokalen Betrachtungsraumes durchdringungsfrei sind, können sie an einer anderen Stelle kollidieren. Durch die Kombination mit einer Kollisionserkennung auf dem triangulierten Modell kann der Benutzer auf solche Situationen hingewiesen werden.

- Durchdringungsprüfung

  Die oben beschriebene Durchdringungsprüfung trifft als Annahme planare Nachbarflächen, die senkrecht auf dem Attribut ’Mittelachse’ stehen. Möchte man diese Randbedingungen fallenlassen, so ist eine Durchdringungsprüfung auf der Basis der genannten Attributinformation nicht mehr möglich; eine Erweiterung der Attributinformation wird notwendig. Handelt es sich bei den Nachbarflächen um Freiformflächen, so ist die Durchdringungsprüfung rechenzeitaufwendig und nicht mehr in Echtzeit möglich.

- Geometrieänderungen des Attribut-tragenden topologischen Elementes

  Während des Modellierungsprozesses können Modifikationen an der Geometrie des Attribut-tragenden topologischen Elementes auftreten, z.B. wird ein Bolzen mit einer Nut versehen. Dies führt zur Änderung des Geometrie der Mantelfläche des Bolzens, die bislang

das Attribut trug. In solchen Situationen kann ein Attribut-basierter Ansatz an seine Grenzen gelangen, da die Merkmale der resultierenden Fläche kaum mehr sinnvoll in Form von Attributen repräsentiert werden können. Eine Möglichkeit ist die Repräsentation als Kontur. Bei der Untersuchung auf korrelierende Attribute müßte dann ein Formvergleich (shape matching [96]) erfolgen. Fraglich ist, ob dies mit hinreichender Geschwindigkeit und Genauigkeit durchgeführt werden kann.

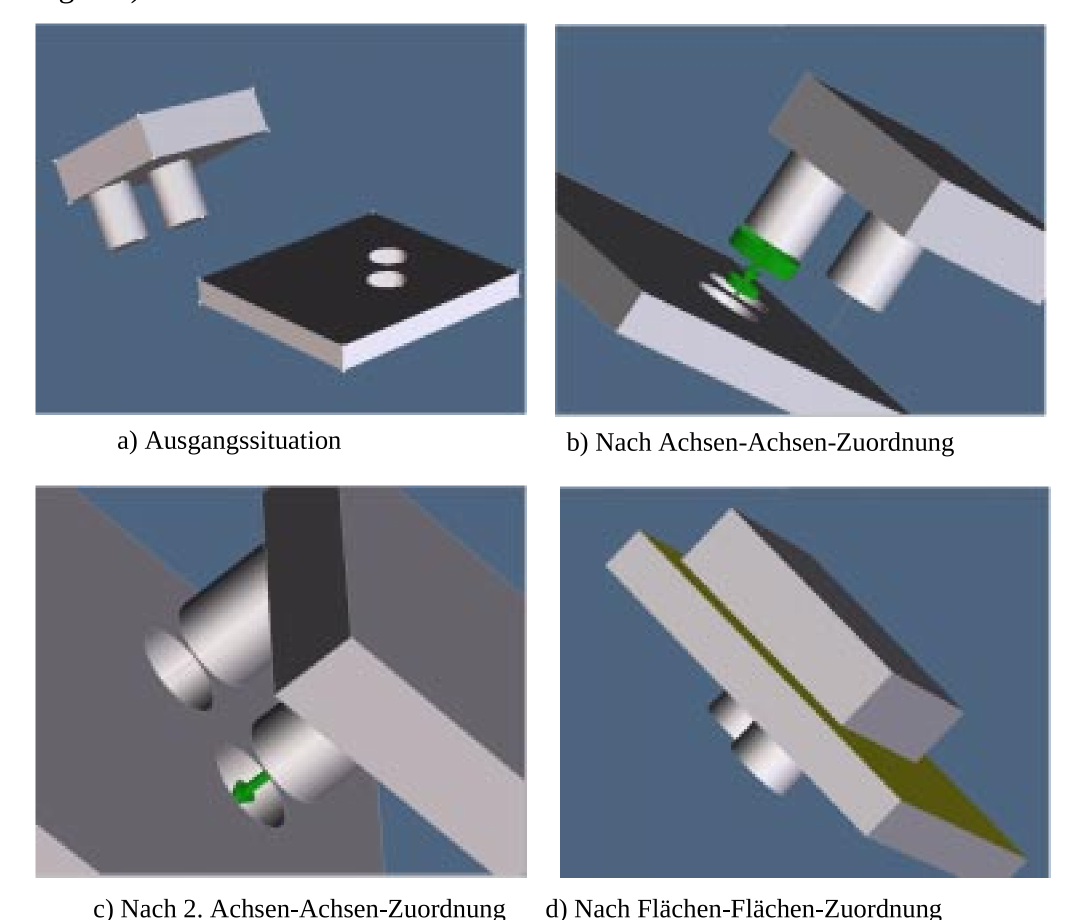
- Bauteile mit mehreren korrelierenden Attributen

  Abbildung 74 zeigt Bauteile mit mehreren korrelierenden Attributen. Für einen solchen Fall ist eine Erweiterung zu realisieren, die es erlaubt, das Verfahren nach einem ersten Montageschritt mit den verbleibenden Bewegungsfreiheitsgraden fortzusetzen, um weitere Attributkorrelationen aufzubauen. Im Rahmen eines diese Arbeit begleitenden Forschungs- und Entwicklungsprojektes erfolgte eine Adaption der hier vorgestellten Ideen in ein kommerzielles CAD-System. Die Umsetzung enthält eine derartige Erweiterung und unterscheidet sich von dem hier vorgestellten Verfahren durch folgende Punkte:

  - Verwendung eines 2D-Eingabegeräts.
  - Explizite Zuordnung korrellierender Attribute (Flächen, Kanten/Achsen, Ecken) durch den Benutzer.
  - Visualisierung der verbleibenden Freiheitsgrade nach dem Vorbild der *center shapes*.

  Das sukzessive Zuordnen korrellierender Attribute wird im folgenden an einem Beispiel verdeutlicht. Dabei wird in eine Platte mit 2 Bohrungen ein ‚Gegenstück‘ eingebracht (siehe Abbildung 84a), in dem zunächst eine Achsen-Achsen-Zuordnung erfolgt. Nach Zuordnung der Achsen durch den Benutzer wird das zu montierende Bauteil so positioniert und orientiert, daß die beiden gewählten Achsen koinzident sind (siehe Abbildung 84b). Ein Visualisierungselement stellt die verbleibenden Freiheitsgrade (Rotation und Translation um die bzw. entlang der Achse) dar. In einem zweiten Schritt werden vom Benutzer die beiden verbleibenden Achsen zugeordnet. Das Bauteil wird gemäß der verbleibenden Freiheitsgrade um die erste Achse rotiert, so daß die beiden neuerlich zugeordneten Achsen koinzident liegen (siehe Abbildung 84c). Position und Gestalt des visuellen Feedbacks werden angepaßt und zeigen dem Benutzer, daß nur noch ein Translationsfreiheitsgrad

verbleibt. Eine letzte Flächen-Flächen-Zuordnung bringt das Bauteil in seine Endlage (siehe Abbildung 84d).



a) Ausgangssituation

b) Nach Achsen-Achsen-Zuordnung

c) Nach 2. Achsen-Achsen-Zuordnung

d) Nach Flächen-Flächen-Zuordnung

Abbildung 84: Zusammenbausimulation durch sukzessives Zuordnen korrellierender Attribute

Aus der Diskussion geht hervor, daß der vorgestellte Ansatz prinzipielle Beschränkungen aufweist, aber gegenüber den aus der Literatur bekannten Verfahren zur Zusammenbausimulation auch grundlegende Vorteile bietet:

- Das Arbeiten mit mathematisch exakter Information.
- Keine Modellierung von Assembly-Features durch den Benutzer.
- Einseitig beschränkte Freiheitsgrade.

Die Beschränkungen hinsichtlich der Komplexität der Geometrie der Objekte sind in der Praxis oft weniger relevant als es den Anschein haben mag, da Verbindungen zwischen Objekten häufig geometrisch einfach ausgeprägt sind; sonst wären Assembly Features ebenfalls inadäquat [127]. Für die Erweiterung des Verfahrens bezüglich einer größeren Bandbreite von Primitiven muß auf weiterführende Arbeiten verwiesen werden, in denen auch eine detailliertere Untersuchung von Implementierungsaufwand, Laufzeitkomplexität und Nutzen des Verfahrens erfolgen sollte. Die Benutzerunterstützung, die die bestehende Implementierung für zylindrische Steckverbindungen bietet, wird in Kapitel 5.1.3 evaluiert.

4.2.12 Skizzieren von Freiform-Flächen am Virtuellen Tisch

Die bislang eingeführten Interaktionstechniken dienen der Modellierung und Assemblierung von Volumenmodellen. Sie sind am herkömmlichen Arbeitsplatz genauso nutzbar wie in einer (semi-)immersiven Umgebung, z.B. am Virtuellen Tisch [160].

Über die Möglichkeiten der 3D-Eingabe mit einem tischgebundenen Eingabegerät am herkömmlichen Arbeitsplatz hinaus, eröffnet der Virtuelle Tisch neue Möglichkeiten insbesondere zur Freiformflächenmodellierung [161]. Aufgrund seiner Größe ist der Virtuelle Tisch für die Freiformflächenmodellierung mittels raumgreifender Handbewegungen geradezu prädestiniert. Bei stereoskopischer Ausgabe scheint das Modell über der Tischoberfläche zu schweben und ermöglicht somit die direkte Interaktion im Raum über der Darstellungsfläche. Der im Rahmen dieser Arbeit eingesetzte Virtuelle Tisch (Barco Baron) kommt mit einer Bildschrimdiagonale von 1,7 Metern dem Aktionsradius des menschlichen Arms sehr nahe.

Ziele bei der Entwicklung von Interaktionstechniken zum Skizzieren von Freifromflächen sind:

- Schnelles, intuitives (aber unpräzises) Erstellen von Freiformflächen durch Handbewegungen.
- Unterstützung bekannter und akzeptierter Konzepte zur Definition von Freiformflächen, wie z.B. Coons-Patches und Skin-Surfaces.
- Visuelle Rückkopplung während der Erzeugung.

Anders als mit den bisher vorgestellten Interaktionstechniken wird damit mehr der konzeptionelle Entwurf addresiert.

Die Intuitivität der Interaktionen soll erreicht werden, in dem der Benutzer von dem Wissen über die mathematischen Grundlagen der Freiformflächen entbunden wird. Er soll an den Virtuellen Tisch treten und Freiformflächen durch Handbewegungen skizzieren können, ohne sich über Kontrollpunkte, Knotenvektoren, u- und v-Richtung etc. Gedanken machen zu müssen. Die Algorithmen zur Interpolation/Approximation von Freiformflächen, wie z.B. Coons-Patches, setzen allerdings bestimmte Eingangsparameter voraus. Die Interaktionstechniken sollen also einerseits dem Benutzer Intuitivität und Flexibilität bieten, müssen aber auch in der Lage sein, aus den Handbewegungen des Benutzers die für die Erzeugung von Freiforflächen nötigen Parameter zu extrahieren (siehe Abbildung 85). Die Interpolation/Approximation der Freiformflächen erfolgt durch den Modellierkern ACIS.

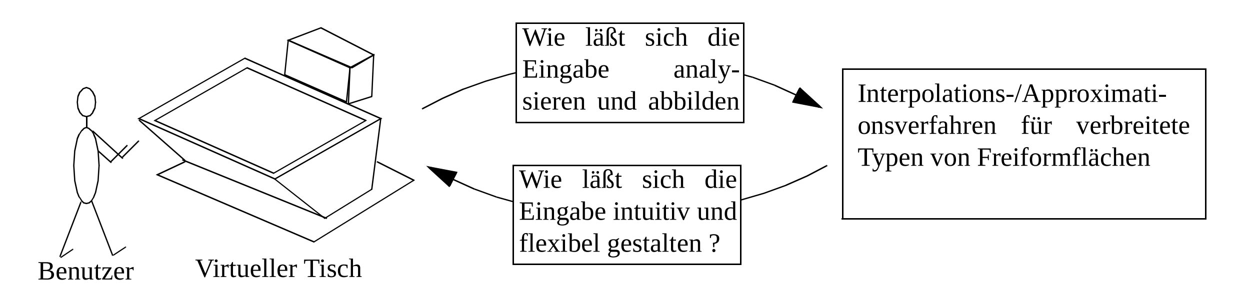
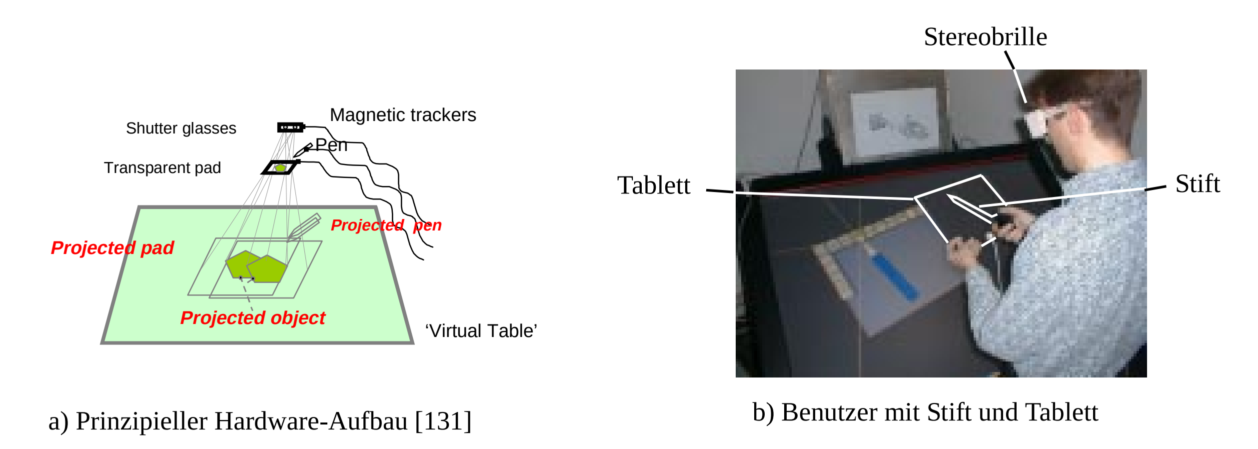


Abbildung 85: Problemstellung der intuitiven Freiformflächenmodellierung am Virtuellen Tisch

Als Interaktions-Front-End kommt die Toolbox Studierstube/VT [131] zum Einsatz. Studierstube/VT bietet Interaktionsbasisfunktionalität, um mit einem transparenten Tablett und einem Stift an einem Virtuellen Tisch zwei-händig interagieren zu können. Die Position und Orientierung beider

Eingabegeräte sowie die des Kopfes des Benutzers wird mit Magnettrackern erfaßt (siehe Abbildung 86a). Das transparente Tablett bietet u.a. die Möglichkeit, 3D-Bedienelemente, z.B. Menüs, in die Szene einzublenden, mit denen mit dem Stift interagiert werden kann (siehe Abbildung 86b). Studierstube/VT bietet als allgemeines Interaktionstoolkit keine Modellierungsfunktionalität.



a) Prinzipieller Hardware-Aufbau [131]

b) Benutzer mit Stift und Tablett

Abbildung 86: Benutzer am Virtuellen Tisch mit Stift und Tablett

Im folgenden werden unterschiedliche Interaktionstechniken zur Generierung von Freiformflächen beschrieben. Die Ansätze zur Generierung von Freiformflächen führen die Ideen von 3-Draw [126] fort und folgen dem Prinzip der Erzeugung charakteristischer Kurven zur Definition von Freiformflächen [60]. Im Unterschied zu 3-Draw werden aus den Interaktionen Freiformflächen abgeleitet; mit 3-Draw können nur Kurven generiert werden. Die semi-immersive Umgebung erleichtert im Vergleich zu 3-Draw die Hand-Auge-Koordination, da am Virtuellen Tisch direkt in 3D interagiert wird und die Position der virtuellen Umgebung mit der der Hand weitgehend übereinstimmt. Schkolne und Schröder präsentieren in [130] ihren Surface-Drawing-Ansatz, bei dem der Benutzer mit Hilfe eines Datenhandschuhs Polygonnetze in 3D 'malt'. Damit unterscheidet sich ihr Ansatz grundlegend von den hier präsentierten Verfahren dadurch, daß keine Freiformflächen im eigentlichen Sinn entstehen. Von dem von Dani et al. [45] vorgestellten Ansatz unterscheiden sich die hier entwickelten Interaktionstechniken, in dem nicht mit einer Default-Fläche begonnen wird, deren Kontrollpunkte durch Selektion und 3D-Interaktion verschoben werden, sondern dem Benutzer die Möglichkeit gegeben wird, direkt im freien Raum mit der Erzeugung von Freiformfläcen zu beginnen. Die Manipulation von Kontrollpunkten wird heute als wenig intuitiv beurteilt [75], da:

- die gewünschte Form schwer zu erzielen ist,
- Benutzer, die mit den mathematischen Konzepten von Freiformflächen nicht vertraut sind, die Wechselwirkung zwischen Kontrollpunkten und Flächenform nicht verstehen,
- die Kontrollpunkte durch das zu manipulierende Objekt verdeckt werden können oder die Darstellung der Kontrollpunkte - ähnlich wie 3D-Widgets - die Szene überfrachtet.

4.2.12.1 Coons-Flächen aus einer 3D-Kontur

Coons-Flächen [40] sind durch zwei Paar Konturkurven definiert; je eins in u- bzw. v-Richtung. Um eine Coons-Fläche durch die Kurvenpaare interpolieren zu können, müssen die Kurvenpaare kompatibel gemacht werden, so daß sie vom selben Grad und über den selben Knotenvektor definiert sind [114]. Möchte der Benutzer mit einem herkömmlichen System eine Coons-Fläche erzeugen, muß er vier Kurven erzeugen und spezifizieren, welche Kurven zu welcher Richtung (u

oder v) gehören (ggfs. müssen die Kurven eine bestimmte Richtung haben). Ein nachfolgender Interpolationsschritt generiert dann die Coons-Fläche.

Zum schnellen Skizzieren einer Coons-Fläche wäre es für den Benutzer viel angenehmer, könnte der die Fläche durch eine geschlossene äußere 3D-Kontur definieren. Er müsste sich nicht um die Zuweisung der Kurven zu der u- und v-Richtung kümmern und könnte dies dem System überlassen. Die Schwierigkeit liegt dabei darin, die Kontur so aufzutrennen, daß die entstehende Coons-Fläche den Erwartungen des Benutzers entspricht. Zu diesem Zweck wurden drei Verfahren implementiert und evaluiert [129]:

- Äquidistante Trennung der Kontur im Parameterbereich.
- Finden der Konturtrennpunkte anhand der Bounding-Box um die Konturpunkte im Weltkoordinatensystem.
- Finden der Konturtrennpunkte anhand einer minimierten Bounding-Box um die Konturpunkte.

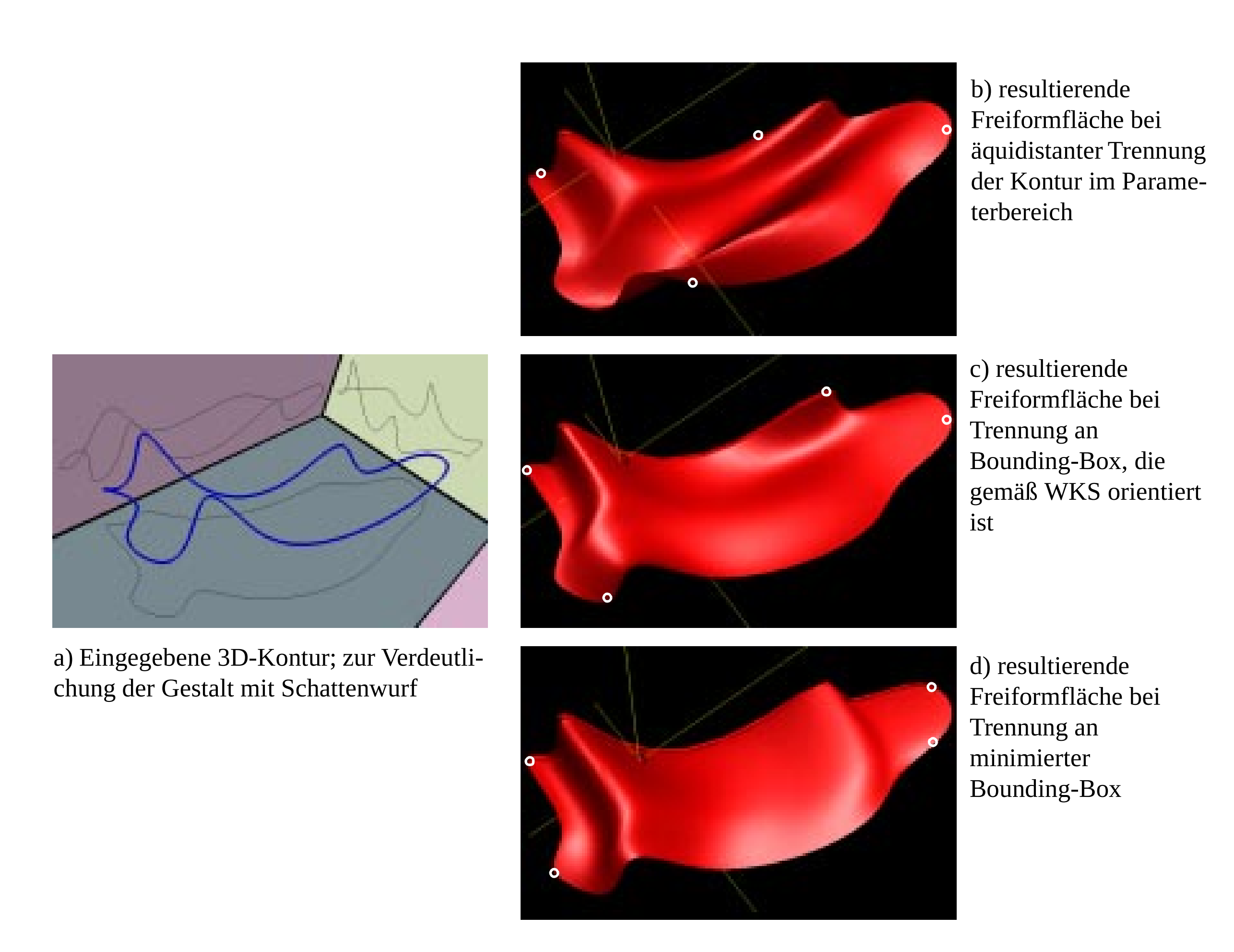


Abbildung 87: Auswirkung der Wahl der Konturtrennpunkte auf die resultierende Coons-Fläche [129]

Abbildung 87 zeigt die Resultate der drei Alternativen anhand eines Beispiels. In Abbildung 87a ist die Konturkurve mit Schattenwurf dargestellt, um die räumliche Gestalt der Kurve zu verdeutlichen. Abbildung 87b zeigt die Coons-Fläche nach äquidistanter Trennung der Kontur im Parameterbereich. Dieses Verfahren zeigt ein unbefriedigendes Ergebnis und ist mit dem Problem behaftet, vom Startpunkt der Kontur abhängige Resultate zu liefern, d.h. abhängig davon, wo der Benutzer beginnt, die Kontur zu skizzieren, liefert das Verfahren bei gleicher Gestalt der Kontur

unterschiedliche Ergebnisse. Das zweite Verfahren bestimmt eine am Weltkoordinatensystem orientierte Bounding Box um die Kontur. Zu den vier kürzesten Kanten der Bounding Box wird jeweils der nächste Punkt der 3D-Kontur bestimmt. Diese Punkte trennen die Kontur. Abbildung 87c zeigt das entsprechnde Ergebnis. Symmetrien in der Kontur werden von der resultierenden Fläche besser nachvollzogen und die Fläche ist weniger wellig. Im Gegensatz zum ersten Verfahren ist dieses invariant gegenüber dem Startpunkt der Kontur, allerdings ist es nicht unabhängig von der Orientierung der Kontur im Raum. Das dritte Verfahren, dessen Ergebnis in Abbildung 87d gezeigt ist, minimiert die Größe der Bounding Box in einem iterativen Verfahren, um eine möglichst kleine Box um die Kontur zu bestimmen [129]. Das Verfahren kommt einem *least square fit* nahe, mit dem ebenfalls zunächst eine Regressionsebene mit minimalen Fehlerquadraten durch den Raum der Konturpunkte gelegt werden könnte, um anschließend die Punkte auf diese Ebene zu projizieren und daraus eine Bounding-Box zu bestimmen.

Beim Skizzieren von Coons-Flächen liefert das dritte Verfahren befriedigende Ergebnisse. 'Berge' und 'Täler' in der Kontur setzen sich entlang der 'Vorzugsrichtung' fort. Auch ist die Coons-Fläche noch glatter als bei Verfahren zwei, was einem energetisch günstigeren Zustand und somit eher dem Verhalten von deformierbaren, physikalischen Stoffen entspricht. Außerdem ist es robuster gegenüber der Orientierung der Kontur im Raum als das zweite Verfahren. Es gibt aber durchaus Situationen, in denen die automatisch gefundenen Konturtrennpunkte und die daraus resultierende Coons-Fläche nicht mit der Intention und Erwartung des Benutzers übereinstimmt. In solchen Fällen könnte auf eine manuelle Definition der Trennpunkte nach Eingabe der Kontur zurückgegriffen werden.

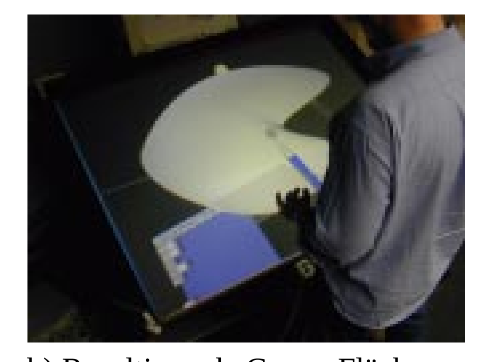
Abschließend sei die Erzeugung einer Coons-Fläche aus einer 3D-Kontur aus Benutzersicht dargestellt:

- Der Benutzer wählt den 3D-Button 'Coons-Fläche' auf der Palette aus.
- Er bewegt den Stift über dem Tisch an die gewünschte Startposition.
- Er setzt den Startpunkt und beginnt in 3D zu zeichnen.
- Das System stellt als graphische Echo eine geschlossene Polylinie drei-dimensional im Raum dar, wodurch der Benutzer einen - wenn auch groben - Eindruck von der Gestalt der späteren Fläche erhält (siehe Abbildung 88a).
- Mit einem erneuten Klick eines der Buttons des Stifts, schließt der Benutzer die Aktion ab.

Jetzt approximiert das System die erfaßten 3D-Punkte mit einer Kurve, wobei der Approximationsfehler einstellbar ist, trennt die Kurve auf und übergibt die Kurvenstücke paarweise sortiert an den Algorithmus, der daraus eine Coons-Fläche generiert. Das Ergebnis wird als schattierte Fläche dargestellt (siehe Abbildung 88b).



a) Coons-Fläche in Erzeugung



b) Resultierende Coons-Fläche

Abbildung 88: Erzeugung einer Coons-Fläche durch eine 3D-Kontur aus Benutzersicht

4.2.12.2 Skin-Flächen mit unmittelbarer visueller Rückkopplung

Die Flächenerzeugung durch *skinning* ist ein anderes, häufig genutztes Konzept in der Freiformflächenmodellierung. Dabei wird eine Menge von Kurven durch eine Fläche interpoliert, die sich unter Wahrung bestimmter Kontinuitätsbedingungen wie eine Haut (skin) an die Kurven anschmiegt [114].

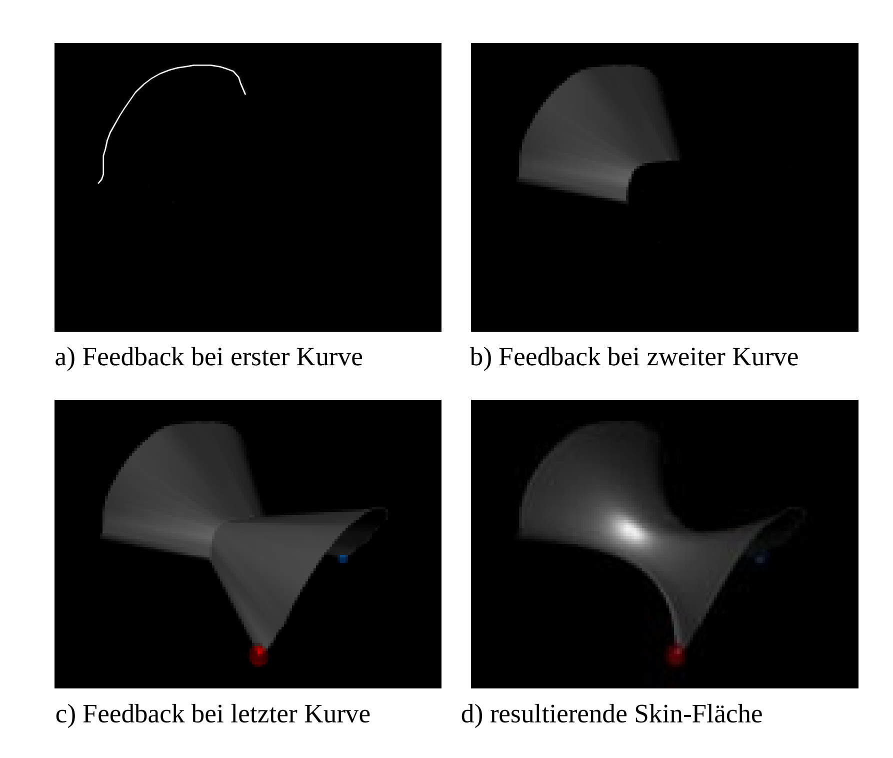
Bei der üblichen Interaktionsprozedur muß der Benutzer sukzessive mehrere Kurven eingeben und dann die Interpolation anstossen. Nachdem die Berechnung der Skin-Fläche durchgeführt ist, wird das Resultat dargestellt. Während der Definition der Kurven bekommt er keine Vorschau auf die resultierende Fläche. Da die Interpolation zu rechenzeitaufwendig ist, um in jedem Interaktionsschritt ausgeführt werden zu können, gilt es ein Verfahren zu finden, das dem Benutzer die Fläche - wenn auch angenähert - im Interaktionsprozeß darstellt, um ihm eine bessere Kontrolle über das Ergebnis zu geben und die entstehende Fläche besser vorhersehen zu können.

Durch adaptive Vermaschung der Kurven kann eine Vorschau auf die resultierende Skin-Fläche, schon während jede einzelne Kurve interaktiv erzeugt wird, in Echtzeit gegeben werden. Dabei entsteht eine sukzessiv wachsende Menge vermaschter Dreiecke als visuelle Rückkopplung. Im Gegensatz zur Interpolation der Kurven durch eine Freiformfläche, können Dreiecksnetze im Interaktionsprozeß schnell erzeugt und dargestellt werden. Sie geben dem Benutzer eine approximative, aber dennoch aufschlußreiche, visuelle Vorschau auf die resultierende Skin-Fläche schon während der Erzeugung der Kurven.

Eine Gegenüberstellung der Aktionen des Benutzers und der internen Abläufe der Interaktion soll den Ablauf verdeutlichen.

Tabelle 7: Ablauf der Skizzierung von Skin-Flächen

Aktionen des Benutzers	Verarbeitung durch Interaktionstechnik
Skizzieren der ersten Kurve (Polylinie) im Raum.	Die vom 3D-Tracker gelieferten Raumpositionen werden mit Hilfe einer Polylinie verbunden und dargestellt (siehe Abbildung 89a).
Skizzieren einer weiteren Kurve (Polylinie); diese Aktion kann wiederholt ausgeführt werden (siehe Abbildung 89c).	Entlang des Verlaufs der Polylinie werden in jedem Interaktionsschritt soviele Punkte eingeführt wie nötig sind, um diese mit ihrer Vorgängerin vermaschen zu können. Zwischen den Polylinien wird eine Menge ebener Flächen dargestellt (siehe Abbildung 89b).
Anstossen der Interpolation nachdem die letzte zur Definition der Skin-Fläche benötigte Kurve (Polylinie) skizziert ist.	Die Polylinien werden durch Kurven approximiert. Durch die Kurven wird eine Skin-Fläche gelegt. Das Ergebnis wird visualisiert (siehe Abbildung 89d).



a) Feedback bei erster Kurve b) Feedback bei zweiter Kurve

c) Feedback bei letzter Kurve d) resultierende Skin-Fläche

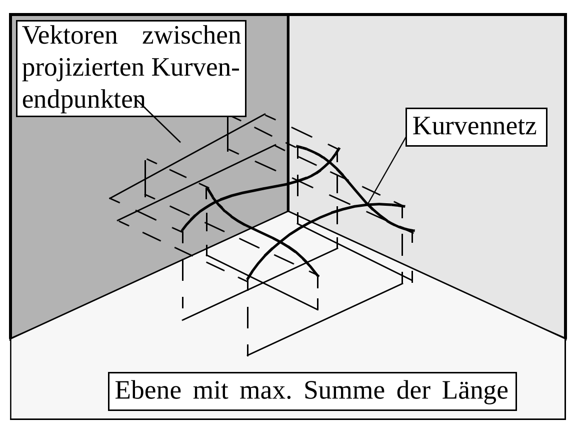
Abbildung 89: Erzeugung einer Skin-Fläche aus Benutzersicht

4.2.12.3 Netz-Flächen

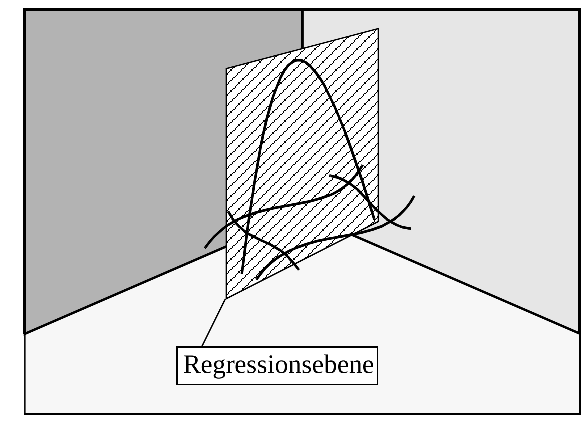
Netz-Flächen (net-surfaces) werden durch ein Netz von Kurven definiert. Sie können als Erweiterung von Coons-Flächen betrachtet werden und erlauben eine bessere Kontrolle über die innere Form einer Freiformfläche. Die von ACIS zur Verfügung gestellten Approximationsalgorithmen für Kurvennetze setzen nicht voraus, daß die Kurven entlang der einen Richtung die der anderen Richtung schneiden - wie Gordon-Flächen [70] dies tun. Damit sind die Verfahren gut für Skizzierungstechniken geeignet, da in einem graphisch-interaktiven Prozeß in 3D eine Berührung der Kurven nicht ohne Weiteres, wie z.B. Snapping, gewährleistet werden kann.

Ähnlich wie bei den Coons-Flächen spielt die Richtung, Reihenfolge sowie Zugehörigkeit der Kurven zur u- bzw. v-Richtung eine entscheidende Rolle. Der Benutzer sollte bei der Definition der Kurven jedoch eine größtmögliche Flexibilität haben und die Kurven in beliebiger Richtung und Reihenfolge kreieren können. Dies wird ermöglicht, in dem die Kurven analysiert, klassifiziert und geordnet werden, bevor sie dem Approximationsalgorithmus zugeführt werden. Dazu werden vier Schritte ausgeführt:

- Projizieren der Endpunkte jeder Kurve auf die drei Hauptebenen des Weltkoordinatensystems.
- Wahl der Ebene so, daß die Summe der Längen der durch die projizierten Endpunkte definierten Vektoren maximal ist (siehe Abbildung 90a).
  Dadurch wird erreicht, daß für die anschließende Klassifikation und Sortierung die Ebene benutzt wird, ’über’ der das Netz definiert ist. Zu beachten ist, daß in diesem Verfahrensschritt eine Regressionsebene durch die Puktemenge der Kurven als Lösungsansatz nicht in Frage kommt, da sie bei Flächen mit im Verhältnis zur ’Grundfläche’ starker Konkavität zu einem falschen Ergebnis führen würde (siehe Abbildung 90b).
- Klassifizieren der Kurven in zwei Hauptrichtungen anhand der Richtung der Vektoren.
- Sortieren der Kurven entlang der Hauptrichtungen (aufsteigend nach Mittelpunkt der Vektoren).



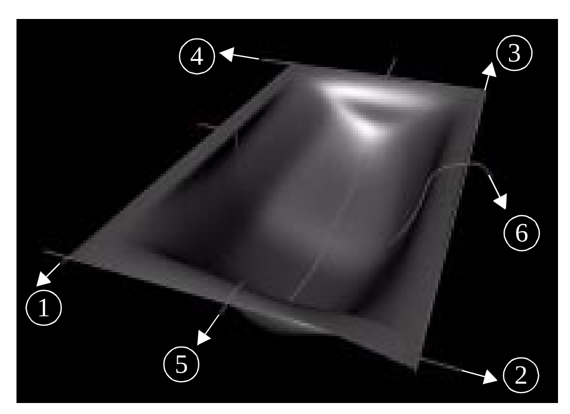
a) Wahl einer geeigneten Ebene durch Projektion der Kurvenendpunkte



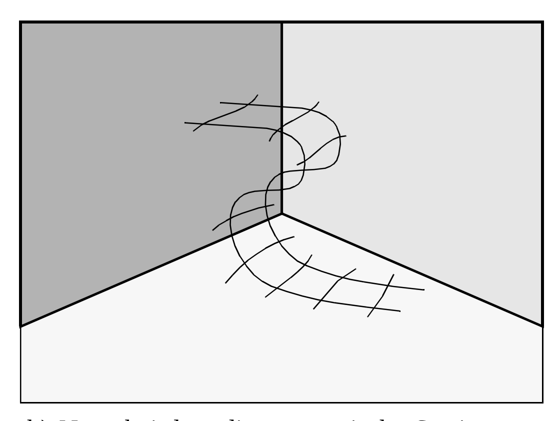
b) Ungeeignetes Vorgehen mit Regressionsebene

Abbildung 90: Auswahl der Ebene zur Klassifikation der Netz-Kurven

Durch dieses Vorgehen ist der Benutzer beim Skizzieren der Kurven des Netzes bezüglich Reihenfolge und Richtung flexibel (siehe Abbildung 91a). Die automatische Sortierung kann allerdings auch der Intention des Benutzers widersprechen, möchte er Hinterschneidungen erzielen (siehe Abbildung 91b).



a) Netz-Fläche mit definierenden Kurven



b) Netz, bei dem die automatische Sortierung zu unerwünschtem Resultat führt

Abbildung 91: Beispiel einer möglichen und einer ’unmöglichen’ Netz-Fläche

Abbildung 92 zeigt die Skizze einer Außenhaut eines Automobils bestehend aus einer Netzfläche.

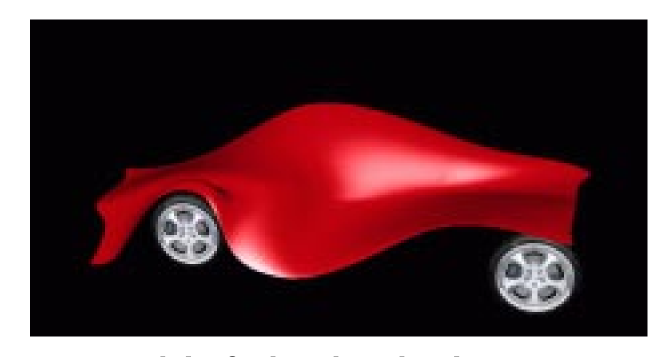
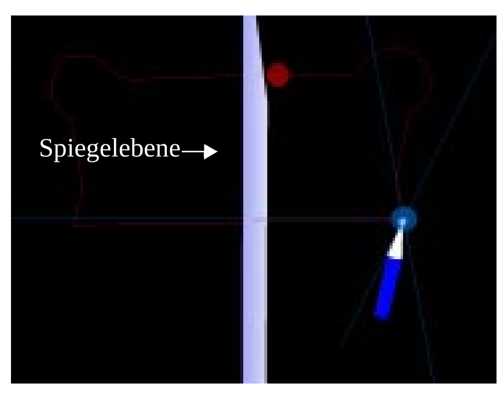


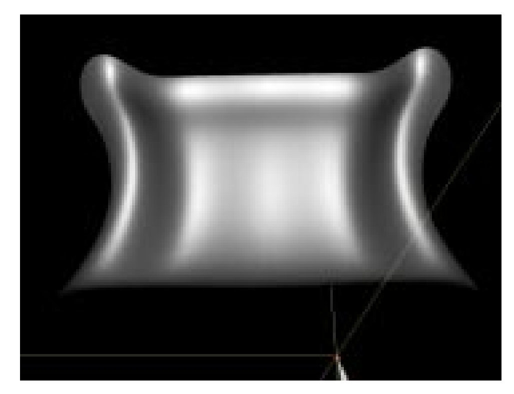
Abbildung 92: Skizze einer Automobilaußenhaut bestehend aus einer Netz-Fläche

4.2.12.4 Symmetrische Freiform-Flächen durch Nutzung der Palette als virtuellen Spiegel

Industrielle Produkte weisen oft Symmetrien auf, z.B. ist eine Karosserie zum größten Teil symmetrisch. Diese Tatsache führt soweit, daß in der Automobilindustrie symmetrische Teile nur zur Hälfte konstruiert werden; nur die nicht-symmetrischen Teile werden vollständig modelliert[8]. Mit herkömmlichen CAD-Systemen wird zunächst eine Hälfte der symmetrischen Fläche erzeugt und anschließend gespiegelt. Dabei ist auf Kontinuitätsbedingungen am Übergang der beiden Teilflächen an der Spiegelebene zu achten.



a) Graphisches Feedback während der Erzeugung einer sym. Coons-Fläche



b) Resultierende sym. Coons-Fläche

Abbildung 93: Symmerische Coons-Fläche

Unter Nutzung von Tablett und Stift kann eine 2-händige Interaktionstechnik zur Verfügung gestellt werden, die die Definition symmetrischer Flächen im Interaktionsprozeß ermöglcht. Dabei wird mittels des in der nicht-dominanten Hand geführten Tabletts die Spiegelebene definiert; eine semi-transparent Fläche visualisiert Position und Orientierung der Spiegelebene (siehe Abbildung 93a). Mit dem ersten Punkt der flächen-definierenden Kurve(n) wird die Spiegelebene im Raum fixiert. Während der Interaktion werden alle vom Tracker gelieferten Positionen an der Spiegelebene gespiegelt und ein entsprechendes symmetrisches visuelles Feedback erzeugt (siehe Abbildung 93a). Nach Abschluß der Eingabe werden die vom Tracker gelieferten sowie die gespiegelten Positionen dem Interpolationsalgorithmus zugeführt - es entsteht eine symmetrische Freiformfläche (siehe Abbildung 93b).

Die Symmetrierung wurde sowohl für Coons- als auch für Skin-Flächen implementiert. Bei den Coons-Flächen wird beim Bestimmen der Kurventrennpunkte die Orientierung der Spiegelebene miteinbezogen. Für Netz-Flächen ließe sich eine Spiegelung ebenfalls realisieren, allerdings müsste analysiert werden, ob die gerade eingegebene Kurve quasi orthogonal oder parallel zur Spiegelachse steht, um ein korrektes visuellen Feedback erzeugen zu können.

4.2.12.5 Diskussion

Mit den eingeführten Interationstechniken ist es dem Benutzer möglich, Freiformflächen durch Skizzieren von flächendefinierenden Kurven direkt in 3D zu erzeugen. Die Interaktionstechniken werten die Eingabedaten automatisch aus und entbinden den Benutzer somit von der Einhaltung der Anforderungen der Interpolationsalgorithmen bzw. dem genauen Wissen über die mathemati-

8. Durch den Trend zum virtuellen Prototypen wird zunehmend dazu übergegangen auch den symmetrischen Teil des Modells vorzuhalten. Die durch Spiegelung erzeugte ’andere Hälfte’ des virtuellen Automobils führt bei der Aufbereitung der CAD-Daten für VR zu Problemen, da von den CAD-Systemen mit der Spiegelung häufig auch die Flächennormale ’gedreht’ wird.

schen Grundlagen von Freiformflächen. Damit wird dem Benutzer ein gesteigertes Maß an Flexibilität geboten; er muß sein Vorgehen bei der Definition von Freiformflächen nicht strikt planen. Ergänzt werden die Interaktionstechniken durch Visualisierungstechniken, die dem Benutzer einen Eindruck von der entstehenden Freiformfläche geben, in dem sie im Interaktionsprozeß eine angenäherte Darstellung erzeugen.

Im Rahmen dieser Arbeit wurde zur Manipulation bzw. Modifikation auch mit direkter Flächendeformation und Trimming experimentiert. Zur Flächendeformation kam das von ACIS zur Verfügung gestellte Verfahren zum Einsatz. ACIS erlaubt es, die Position von Punkten auf der Fläche direkt zu modifizieren und berechnet mit Hilfe eines Optimierungsverfahrens die Fläche neu, so daß die in der Fläche repräsentierte Energie minimal ist. Dieses aufwendige Verfahren ist als *varational surface design* bekannt und wird von Welch und Witkin in [177] beschrieben. Für einen interaktiven Prozeß, bei dem in jedem Schritt die Position eines Punktes oder einer Punktemenge der Oberfläche geändert wird, hat sich das Verfahren für komplexe Freiformflächen als noch nicht hinreichend schnell erwiesen [129]. Diese Einschränkung wird durch die weiterhin zunehmende Rechengeschwindigkeit stetig kompensiert, so daß die im Vergleich zur Kontrollpunktmanipulation intuitivere direkte Freiformflächendeformation künftig auch im direktmanipulativen Prozeß eingesetzt werden können wird. In Kombination mit Tablett und Stift entstehen neue Möglichkeiten zur intuitiven Selektion und Modifikation von Bereichen einer Freiformfläche, z.B. könnte das Tablett dazu benutzt werden, um eine zu modifizierende Kurve - als Schnittkurve zwischen Freiformfläche und Tablett - zu bestimmen. Für diese und andere Ideen zur Freiformflächenmanipulation am Virtuellen Tisch sei auf weiterführende Arbeiten am Ende dieser Arbeit verwiesen.

4.3 Zusammenfassung

In diesem Kapitel wurden neuartige 3D-Interaktionstechniken für 3D-Eingabegeräte präsentiert, die die schnelle und gleichzeitig präzise Modellerzeugung und -manipulation ermöglichen. Kombiniert wurden diese Interaktionstechniken mit visuellen Hinweisen, die die Orientierung und Navigation erleichtern und dem Benutzer die im Gange befindliche Interaktion oder auch die Modifikationsmöglichkeiten im 3-Dimensionalen anzeigen. Die Visualisierung unterstützt somit die Interaktion in synergetischer Art und Weise.

Bei der Entwicklung von geeigneten Interaktionstechniken für 3D-Eingabegeräte im Kontext 3D-Modellierung wurden die speziellen Anforderungen von CAD, aber auch die motorischen Fähigkeiten des Menschen und die Unzulänglichkeiten heutiger 3D-Eingabegerätetechnologie beachtet. Präzises und gleichzeitig effizientes wie intuitives Arbeiten wird durch diskretisierbare Interaktionstechniken weitestgehend unabhängig von der Präzision des verwendeten 3D-Eingabegerätes realisiert. Damit konnten die Beschränkungen im bisherigen Einsatz von 3D-Eingabegeräten auf das unpräzise Entwerfen im konzeptionellen Design überwunden werden [154].

Als neuartiges Interaktionsparadigma nimmt die Topologie-basierte eingeschränke Modifikationstechnik (TCBM) eine zentrale Rolle im Rahmen der Entwicklungen ein. Die TCBM nutzt die bei 3D-Eingabegeräten vorhandenen Freiheitsgrade auf vorteilhafte Weise, um aus einem Gestenerkennungsschritt Modifikationen mit reduzierter Anzahl von Freiheitsgraden abzuleiten, wie sie im CAD typischerweise benötigt werden. Die TCBM wahrt dabei die Reiz-Reaktions-Korrespondenz in 3D und ermöglicht die direkte Manipulation von CAD-Modellen mittels intuitiver Interaktionen. Das kontext-abhängige Erkennen von Gesten zur Parameteränderung, Translation bzw. Rotation minimiert die Anzahl von Moduswechseln. Die 2-händige TCBM unterstützt beidhändiges Arbeiten im Modellierungsprozeß und ermöglicht die Parallelisierung von Interaktions-

schritten. Die Parametrisierung der TCBM stellt einen Ansatz dar, um Einschränkungen für bestimmte Interaktion, die z.B. aus der Semantik eines Feature-basierten Modells herrühren können, auf der Ebene der Interaktionskomponente zu handhaben und somit nur semantisch-korrekte Modifikationen zu erlauben.

Durch den Einsatz von 3D-Eingabegeräten erschliessen sich neue Möglichkeiten, Modellierungsoperationen zu vereinfachen und durch entsprechende Interaktions- und Visualisierungstechniken direkt-manipulativ zu gestaltet. Als Beispiele sind direkt-manipulatives Sweeping und implizite Boolsche Operationen zu nennen.

Von grundlegender Bedeutung hat sich ein echtes, effizientes und präzises Picking und Snapping in 3D erwiesen. Nur so kann direkt mit den Objekten im Konstruktionsraum agiert werden. Zu diesem Zweck wurde ein 3D-Pick-Verfahren entwickelt und evaluiert, daß sich eines *multi-level bounding box checks* bedient und die Kohärenz aufeinanderfolgender Benutzeraktionen nutzt, um unnötige Pick-Berechnungen zu vermeiden. Das Verfahren zeigt in der Praxis ein weitgehend konstantes Laufzeitverhalten und ist schneller als herkömmliche Ray-Picking Algorithmen. Snapping ist auf präzisen, moderat komplexen CAD-Modellen unabhängig von deren Darstellungsgenauigkeit in Echtzeit möglich. Mit dem Snapping wird nicht nur ein Gravitationseffekt auf den 3D-Cursor ausgeübt, sondern es kann auch dazu genutzt werden, um Modelle auf anderen zu verschieben. In Erweiterung der Snapping-Funktionalität, wurde *repulsion* als neue Interaktionstechnik entwickelt. Repulsion sorgt für eine Abstoßung des 3D-Cursors vom Objekt und ermöglicht z.B. die Erzeugung eines Grundkörpers in einer definierten Entfernung zu einem Basismodell unter Beachtung dessen Orientierung.

Darüber hinaus wurde in dem vorliegenden Kapitel ein Attribut-basierter Ansatz zum computerunterstützten Zusammenbau virtueller Modelle vorgestellt, der das Ziel verfolgt, den Benutzer bei der interaktiven Zusammenbausimulation zu assistieren. Motiviert ist dieser Ansatz durch die Tatsache, daß das präzise, interaktive Positionieren von Bauteilen zu einem Assembly ohne Systemunterstützung praktisch unmöglich ist. Die dem Ansatz zugrundeliegende Idee besteht in einer Anreicherung des CAD-Modells mit Attributen, die für den Zusammenbau relevant sind. Die Generierung der Attributinformation und die Handhabung der Attribute während des Modellierungsprozesses geschieht für den Benutzer transparent, womit sich der Ansatz von Assembly-Feature-basierten Verfahren unterscheidet. In der Zusammenbausimulation wird die Attributinformation dazu benutzt, um Teilmodelle automatisch richtig auszurichten und die Bewegungsfreiheitsgrade für weitere Interaktionen einzuschränken. Unter bestimmten Randbedingungen ist eine Erkennung des Erreichens der Einbaulage, Durchdringungsprüfung und Kollisionsaufhebung alleine basierend auf der mathematisch präzisen Attributinformation möglich.

Motiviert durch die neuen Möglichkeiten des Virtuellen Tisches als Interaktionsgerät wurden schließlich Skizzierungstechniken für Freiformflächen präsentiert. Damit wird der konzeptionelle Entwurf angesprochen, in dem Präzision noch nicht die zentrale Rolle spielt, wie dies in der Modellierung bzw. Konstruktion der Fall ist. Durch intelligente Interaktionstechniken, die die Eingabedaten auswerten und in die von den Interpolationsmethoden gewünschte Form bringen, wird der Benutzer in die Lage versetzt, Freiformflächen durch Skizzieren von Kurven direkt im freien Raum zu definieren. Die Interaktionstechniken steigern die Flexibilität bei der Eingabe seitens des Benutzers und entbinden ihn von explizitem Wissen über die Vorgehensweise der Interpolationsalgorithmen zur Freiformflächengenerierung.

In diesem Kapitel spielte VR-Technologie insbesondere im Hinblick auf 3D-Eingabegeräte und Visualisierung in Verbindung mit dem Anwendungskontext CAD und dessen speziellen Anforde-

rungen eine Rolle. Das vorherige Kapitel widmete sich darüber hinaus Aspekten des computerunterstützten kooperativen Arbeitens. An dieser Stelle sei erwähnt, daß die hier beschriebenen Interaktionstechniken in dem System ARCADE auch im kooperativen Betrieb zur Verfügung stehen, wobei direkte Manipulationen in Echtzeit an die Kooperationspartner übertragen werden. Dies vermittelt bzw. steigert das Gefühl von Telepräsenz in einem gemeinsamen, verteilten, virtuellen Konstruktionsraum, in dem simultan modelliert werden kann.

Im folgenden Kapitel werden die eingeführten Interaktionstechniken einer Evaluierung unterzogen, wobei sowohl ein theoretischer Vergleich mit bekannten Ansätzen aus der Forschung sowie ein praktischer Vergleich mit kommerziellen Systemen erfolgt. Darüber hinaus sind die Ergebnisse einiger gezielter Untersuchungen dargestellt, die sich dem Effekt der Visualisierung auf bestimmte Interaktionen widmen.

5 Evaluierung der 3D-Interaktions- und Visualisierungstechniken

Im vorliegenden Kapitel werden die im Rahmen dieser Arbeit entwickelten 3D-Interaktions- und Visualisierungstechniken im Hinblick auf ihre Einsetzbarkeit und Eignung im Modellierungsprozeß evaluiert. Im Vordergrund der Untersuchungen steht die Frage nach der Effizienz der erarbeiteten Methoden und Verfahren bei Einhaltung der Präzisionsanforderungen.

5.1 Evaluierung der 3D-Interaktionstechniken

Das Ziel einer Evaluierung ist es, Untersuchungsobjekte zu bewerten und diese vergleichend gegenüberzustellen. Zu diesem Zweck sind standardisierte Evaluierungsverfahren wünschenswert. Mit STEP-3D (standard evaluation procedure for 3D interaction) [71] existiert ein Ansatz zur Standardisierung von Evaluierungsverfahren für 3D-Interaktionen, der allerdings nur Basis-3D-Interaktionen ohne den Kontext Modellierung betrachtet. In der Modellierung sind 3D-Interaktionen nicht Selbstzweck, sondern dienen der Erzeugung oder Manipulation von Bauteilen oder -gruppen. Um diesem Spezifikum Rechnung zu tragen, wurden basierend auf der Methodologie, die STEP-3D vorschlägt, Evaluierungen durchgeführt, um die eingeführten Verfahren theoretisch und praktisch - im Rahmen von Benutzertests - zu bewerten und mit existierenden Ansätzen zu vergleichen.

Evaluiert werden im folgenden:

- Die Interaktionstechniken zur Objekterzeugung und -modifikation in Kombination mit den direkt-manipulativen Modellierungsoperationen.
- Die Kombination aus 2D- und 3D-Cursorkontrolle mit der 3D-Maus.
- Das Verfahren zur Zusammenbauunterstützung.

5.1.1 Schnelle und präzise Objekterzeugung

Die Evaluierung der Interaktionstechniken zur schnellen und präzisen Objekterzeugung mit 3D-Eingabegeräten gliedert sich in zwei Teile. Zunächst erfolgt ein qualitativer Vergleich mit Ansätzen und Systemen aus der Forschung, basierend auf theoretischen Betrachtungen. Anschließend werden die Ergebnisse eines Benutzertests dargestellt, in dem die *expert performance* ermittelt wurde, d.h. das Maß an Effizienz, welches erfahrene Benutzer mit dem ihnen vertrauten System bei der Bearbeitung einer gegebenen Aufgabe erzielen können. Dieser Test wurde vergleichend mit kommerziellen CAD-Systemen durchgeführt.

5.1.1.1 Qualitativer Vergleich mit anderen Ansätzen aus der Forschung

Die im Stand der Technik aufgeführten verwandten Ansätze aus der Forschung werden im folgenden einem qualitativen, nicht formalisierten Vergleich unterzogen. Der Vergleich basiert auf Veröffentlichungen und Videos sowie z.T. auf eigenen Erfahrungen mit den einzelnen Ansätzen und Systemen. Für den Vergleich wurden sowohl objektive (meßbare) als auch subjektive Kriterien herangezogen. Die Beurteilung der unterschiedlichen Ansätze wird dadurch erschwert, daß für bestimmte Aufgaben jeweils verschiedene Lösungswege existieren. Um die Menge alternativer Lösungswege jedes Ansatzes überschaubar zu halten, erfolgt der Vergleich anhand folgender drei Beispielaktionen:

- Eingabe eines 3D-Punktes.
- Erzeugen eines Quaders.
- Translieren eines Objektes.

Diese Auswahl an Beispielaktionen erhebt keinen Anspruch auf Vollständigkeit, verdeutlicht jedoch die Charakteristika der unterschiedlichen Ansätze.

Zunächst erfolgt ein Überblick über die verglichenen Konzepte und Systeme (vgl. Kapitel 2.2).

1) Konzept der Arbeitsebene (2D-Eingabegerät und 2D-Mauszeiger)

   Eine Arbeitsebene ist eine Ebene im 3D-Raum, auf der der Konstrukteur Objekte konstruiert. Bevor ein Objekt erzeugt werden kann, muß die Arbeitsebene geeignet positioniert und orientiert werden. Mausbewegungen werden auf die Arbeitsebene abgebildet. Bei diesem Ansatz werden *picks* gewöhnlich durch Schnittpunktberechnung zwischen Strahl und Arbeitebene oder vorhandenen Objekten bestimmt.

2) Konzept der multiplen Ansichten (2D-Eingabegerät und 2D-Mauszeiger)

   Bei dem Konzept der multiplen Ansichten wird dem Konstrukteur das Modell simultan aus verschiedenen Ansichten dargestellt, meist die drei Standardansichten plus eine perspektivische Darstellung. Um einen Punkt im Raum festzulegen, muß in zwei Ansichten ein Pick ausgeführt werden. Der erste Pick legt 2 Koordinaten des Punkes, der zweite Pick die dritte Koordinate fest. Gegebenenfalls kann ein Punkt auf der Oberfläche eines existierenden Objektes in der perspektivischen Projektion gewählt werden.

3) Manipulatoren, 3D-Widgets (2D-Eingabegerät und 2D-Mauszeiger)

   Manipulatoren sind graphische Elemente in der Szene, die 2D-Interaktionen auf 3D-Interaktionen abbilden. Manipulatoren können eingesetzt werden, um existierende graphische Elemente, wie Objekte und Punkte, im Raum zu translieren, rotieren und skalieren.

4) Ansatz von Nielson und Olson (2D-Eingabegerät und 3D-Cursor)

   Bei dem Verfahren von Nielson und Olson läßt sich ein 3D-Cursor mit einem 2D-Eingabegerät positionieren. Dazu wird die xy-Ebene, auf der das 2D-Eingabegerät bewegt wird, in Segmente eingeteilt. Bewegungen innerhalb eines bestimmten Segmentes werden als Bewegungen in z-Richtung interpretiert.

5) Umschalten mit Kontroll-Taste (2D-Eingabegerät und 3D-Cursor)

   Eine Kontroll-Taste, z.B. Shift, wird benutzt, um die Abbildungsebene für die Mausbewegungen umzuschalten. Ohne daß die Kontroll-Taste gedrückt ist, werden Mausbewegungen z.B. auf eine xy-Ebene im Raum abgebildet; wenn die Taste gedrückt ist, werden Bewegungen der Maus in x- oder y-Richtung auf die z-Achse des Raumes abgebildet.

6) JDCAD (3D-Eingabegerät und 3D-Cursor)

   JDCAD ist ein direkt-manipulatives CAD-System, das mit einem fliegenden 3D-Eingabegerät und einem 3D-Zeiger arbeitet.

7) SKETCH (2D-Eingabegerät und 2D-Zeiger)

   SKETCH ist ein Skizzier-/Modelliersystem, das aus 2D-Gesten 3D-Geometrien ableitet und Modellieroperationen, wie Boolsche Verschneidungen und Sweeping, bietet. Es ist in erster Linie für das unpräzise Skizzieren von 3D-Modellen gedacht.

8) ARCADE (3D-Eingabegerät und 3D-Zeiger)

   ARCADE ist das im Rahmen dieser Arbeit implementierte System, welches die vorgestellten Konzepte und Verfahren umsetzt.

Der Vergleich basiert auf folgenden Kriterien:

i) Anzahl der Interaktionen

   Dies ist ein objektiv meßbares Kriterium, da die Anzahl der für eine bestimmte Aufgabe benötigten Mausbewegungen und Mausklicks gezählt werden kann. Zu beachten ist, daß die Geschwindigkeit, mit der diese Interaktionen ausgeführt werden können, die Ausführungszeit einer Aufgabe ebenfalls beeinflußt (siehe Kapitel 5.1.2).

ii) Intuitivität

   Im Unterschied zur Anzahl der Interaktionen ist Intuitivität ein subjektives Kriterium. Die von einem Benutzer empfundene Intuitivität hängt von seinen Gewohnheiten, Erfahrungen und seinem Hintergrund ab. Als Maß für die Intuitivität wurde die Reiz-Reaktions-Korrespondenz gewählt, da sich Objekte in der realen Welt gemäß diesem Prinzip verhalten. Die Reiz-Reaktions-Kompatibilität in 3D wurde höher bewertet als eine in 2D auf der Bildebene.

iii) Präzision

   Präzision ist - auf den ersten Blick - ein objektives Kriterium. Allerdings existieren viele Möglichkeiten, um präzises Arbeiten zu ermöglichen: die alpha-numerische Eingabe von Werten, diskretisierte Interaktionen, etc.

   Die mit einem System erzielbare Präzision wurde umso höher bewertet, je feiner sich das graphische Echo durch Skalierung der physischen Bewegung bewegen läßt und je mehr Mechanismen es zur präzisen graphisch-interaktiven Modifikation von Objekten bietet, z.B. durch Grids, Snapping, etc.

iv) Direkte Manipulation

Ob ein Ansatz oder System direkt-manipulativ arbeitet, läßt sich ad hoc entscheiden. Allerdings bieten manche der betrachteten Ansätze direkte Manipulation nur während bestimmter Interaktionen; sie sind nicht generell direkt-manipulativ.

v) Benutzer-kontrollierte Eingabesequenz

Ein immer wieder von Softwareergonomen und Arbeitswissenschaftlern gefordertes Merkmal ist die flexible Handhabung von Eingabesequenzen durch das System [2]. Nicht das System soll die Reihenfolge von Eingabeparametern vorgeben, sondern der Benutzer sollte die Parameter in der für ihn geeigneten Reihenfolge eingeben können.

Tabelle 8 zeigt die Gegenüberstellung der Ansätze und Systeme im Hinblick auf die Beispielinteraktion 'Eingabe eines Punktes im Raum'. Zu beachten ist, daß die acht verglichenen Verfahren sich in der Tabelle thematisch gruppiert in den fünf Spalten wiederfinden. Darüber hinaus ist generell zu erwähnen, daß das per se unpräzise Arbeiten mit 2D-Maus und 2D-Zeiger sich durch die Verwendung von Gitternetzlinien präziser gestalten läßt. 3D-Widgets sind in Tabelle 8 nicht aufgeführt, da sie vorranging dazu dienen, bereits vorhandene Objekte zu manipulieren. Ein denkbares Vorgehen mit Manipulatoren wäre, auf Knopfdruck einen Punkt umgeben von einem Manipulator im Zentrum des sichtbaren Konstruktionsraumes zu erzeugen und diesen per Translation interaktiv an die gewünschte Position zu bewegen (vgl. Tabelle 10).

Tabelle 8: Vergleich von Eingabemetaphern für die Erzeugung eines Punktes im Raum

	2D-Maus und 2D-Zeiger	2D-Maus mit 3D-Cursor	JDCAD (fliegendes 3D-Eingabegerät und 3D-Cursor)	SKETCH (2D-Maus/Stift und 2D-Zeiger)	ARCADE (3D-Maus mit 3D-Cursor)
Punkt im Raum	a) Multiple Ansichten 2 Bewegungen, 2 Picks - unpräzise b) Arbeitsebene 1 Bewegung, 1 Pick (wenn Arbeitsebene bereits festliegt) - unpräzise	a) Nielson-Olson-Ansatz 1 Bewegung, 1 Pick – geringe Intuitivität ⊕ präzise, wenn Abbildung skalierbar b) Kontroll-Taste 2 Bewegungen, 1 Pick, 1 Tastendruck - geringe Intuitivität ⊕ präzise, wenn Abbildung skalierbar	1 oder mehrere Bewegungen, 1 pick (mehrere Bewegungen wenn Trackingbereich verlassen wird) ⊕ intuitiv ⊕/- Präzision geprägt durch Trackingeinheit, Filterung und Abbildung.	nicht direkt eingebbar; nicht Teil des Konzeptes man arbeitet entweder auf der Grundebene oder auf existierenden Objekten - unpräzise	1 3D-Mausbewegung, 1 Pick ⊕ intuitiv ⊕ präzise aufgrund der Skalierbarkeit und des 3D-Gitters

Tabelle 9 stellt die Eingabemetaphern für die Erzeugung eines Quaders gegenüber. Auch hier wurde auf die Darstellung der 3D-Widgets verzichtet, da diese für die Objekterzeugung 'aus dem Nichts' ungeeignet sind. Eine mögliche Vorgehensweise wäre die Erzeugung eines Quaders mit

vordefinierten Abmesungen im Zentrum des sichtbaren Konstruktionsraumes und anschließende Skalierung und Positionierung mit Hilfe von Manipulatoren.

Tabelle 9: Vergleich von Eingabemetaphern für die Erzeugung eines Quaders

	2D-Maus und 2D-Zeiger	2D-Maus mit 3D-Cursor	JDCAD (fliegendes 3D-Eingabegerät und 3D-Cursor)	SKETCH (2D-Maus/Stift und 2D-Zeiger)	ARCADE (3D-Maus mit 3D-Cursor)
Erzeugen eines Quaders	a) Multiple Ansichten 4 Bewegungen, 2 mal 2 Picks in 2 Ansichten - nicht intuitiv - unpräzise - nicht direkt-manipulativ ⊕ Benutzer-kontrollierte Eingabesequenz b) Arbeitsebene 3 Bewegungen, 3 Picks - geringe Intuitivität (Abbildung der Mausbewegungen) - unpräzise ⊕ direkt-manipulativ - vordefinierte Eingabesequenz	a) Nielson-Olson-Ansatz 2 Bewegungen, 2 Picks - geringe Intuitivität (3D-Cursor ist bei diesem Ansatz schwer zu kontrollieren) ⊕ präzise, nur wenn Abbildung skalierbar ⊕ direkt-manipulativ ⊕ Benutzer-kontrollierte Eingabesequenz b) Kontroll-Taste 4 Bewegungen, 2mal Tastendruck, 2 Picks - geringe Intuitivität ⊕ präzise, nur wenn Abbildung skalierbar ⊕ direkt-manipulativ ⊕ Benutzer-kontrollierte Eingabesequenz	2 oder mehr Bewegungen, 2 Picks (mehr als 2 Bewegungen wenn Trackingbereich verlassen wird) ⊕ intuitiv ⊕/– Präzision geprägt durch Tracking, Filterung ⊕ direkt-manipulativ ⊕ Benutzer-kontrollierte Eingabesequenz	1 Bewegung, dann Geste mit mindestens 4 Bewegungen und 6 Picks o intuitiv - unpräzise - nicht direkt-manipulativ ⊕ Benutzer-kontrollierte Eingabesequenz	2 Bewegungen, 2 Picks ⊕ intuitiv ⊕ präzise ⊕ direkt-manipulativ ⊕ Benutzer-kontrollierte Eingabesequenz

Tabelle 10 zeigt die evaluierten Eingabemetaphern hinsichtlich der Objekttranslation im Vergleich.

Tabelle 10: Vergleich von Eingabemetaphern für die Translation eines Objektes

	2D-Maus und 2D-Zeiger	2D-Maus mit 3D-Cursor	JDCAD (fliegendes 3D-Eingabegerät und 3D-Cursor)	SKETCH (2D-Maus/Stift und 2D-Zeiger)	ARCADE (3D-Maus mit 3D-Cursor)
Translation	a) Multiple Ansichten 1 Bewegung, 1 Pick zur Bestimmung des Referenzpunktes, 2 Bewegungen, 2 Picks in verschiedenen Ansichten, um Zielpunkt zu bestimmen (3D Translation) - geringe Intuitivität - unpräzise - nicht direkt-manipulativ b) Arbeitsebene 2 Bewegungen, 2 Picks (Translation in der Ebene) - geringe Intuitivität - unpräzise - nicht direkt-manipulativ c) Manipulatoren 2 Bewegungen, 1 Pick (lineare oder planare Translation) ⊕ intuitiv – unpräzise ⊕ direkt-manipulativ	a) Nielson-Olson-Ansatz 2 Bewegungen, 2 Picks - geringe Intuitivität ⊕ präzise, nur wenn Abbildung skalierbar ⊕ direkte Manipulation ist möglich b) Kontrolltaste 2 Bewegungen, 1 Tastendruck, 1 Pick zur Objektselektion, dann 2 Bewegungen, 1 Tastendruck, 1 Pick zur Bestimmung des Zielpunktes - geringe Intuitivität ⊕ präzise, nur wenn Abbildung skalierbar ⊕ direkt-manipulativ	1 3D-Rotation und 1 Klick für Picking mit 'Strahlcursor', 1 Bewegung, 1 Pick für Translation (mehrere Bewegungen, wenn Trackingbereich verlassen wird) (3D-Translation) ⊕ intuitiv ⊕/– Präzision geprägt durch Tracking, Filterung, etc. ⊕ direkt-manipulativ	verschiedene Möglichkeiten, aber keine direkt-manipulative Translation in 3D: a) 3D Translation nur durch Objekt-zu-Objekt Snapping b) in z-Richtung mittels Schattenskizze c) in einer Ebene d) entlang skizzierter Linie mit dem Verfahren variierende Anzahl von Interaktionen o intuitiv (a-d) – unpräzise (a-d) ⊕ direkt-manipulativ (c, d) – nicht direkt-manipulativ (a, b)	verschiedene direkt-manipulative Möglichkeiten zur Translation: a) 3D Translation: 2 Bewegungen, 2 Picks b) Topologisch-basierte beschränkte Modifikation: 1Bewegung, 1 Pick für Selektion; 1 Bewegung für Geste und Translation, 1 Pick, um Objekt loszulassen c) Beschränkt auf Raumebenen und -achsen: 1 Bewegung, 1 Pick für Selektion; 1 Tastendruck für Aktivierung des Constraints, 1 Bewegung, 1 Pick, um Objekt loszulassen (zusätzlicher Tastendruck, um Constraints zu wechseln) generell gilt: ⊕ intuitiv ⊕ präzise aufgrund der Skalierbarkeit und Diskretisierung ⊕ direkt-manipulativ

Zusammenfassend läßt sich feststellen, daß die Anzahl der Interaktionen bei Verwendung eines 3D-Eingabegerätes geringer ausfällt als bei 2D-Eingabegeräten. Im Vergleich zu JDCAD bietet ARCADE eine bessere Unterstützung beim genauen Konstruieren und trägt somit den Anforderungen von CAD eher Rechnung. Die entwickelten Interaktionstechniken schöpfen ihre Präzision während direkt-manipulativer Vorgänge aus den vielfältigen Diskretisierungsmöglichkeiten, die der Benutzer gezielt kontrollieren kann. Anzumerken bleibt, daß sich fortschrittliche Konzepte, wie implizite Boolsche Operationen, history pick, etc., in obigen Tabellen aufgrund der Art der Vergleichsbeispiele nicht niederschlagen. Diese kommen hingegen bei dem im nächsten Abschnitt beschriebenen Benutzertest zur Geltung.

5.1.1.2 Quantitativer Vergleich mit kommerziellen CAD-Systemen

In dem bisherigen theoretischen Vergleich wurde keine Aussage über die Geschwindigkeit getroffen, mit der eine bestimmte Aufgabe bearbeitet werden kann. Zwar läßt die Anzahl der Interaktionen und die Intuitivität eines Ansatzes ein bestimmtes zeitliches Verhalten erwarten, allerdings beeinflussen auch die Geschwindigkeit, mit der eine Basisinteraktion ausgeführt werden kann, und die Art und Weise wie Modellieroperationen durchgeführt werden müssen, die Bearbeitungszeit einer Aufgabe. Die Erhebung der *expert performance*, d.h. der Bearbeitungszeit, die erfahrene Benutzer für die Lösung einer gegebenen Konstruktionsaufgabe mit dem ihnen vertrauten System benötigen, dient einem quantitativer Vergleich mit kommerziellen CAD-Systemen[1].

Testdurchführung

Als Testpersonen wurden erfahrene Benutzer von kommerziellen CAD-Systemen gewonnen. Die Konstrukteure und Maschinenbauingenieure hatten täglich - und das z.T. schon seit Jahren - Umgang mit ’ihrem’ CAD-System. Als Vergleichssysteme kamen CATIA von Dassault, IDEAS Master Series von SDRC, Pro/Engineer von PTC und Solid von strässle zum Einsatz.

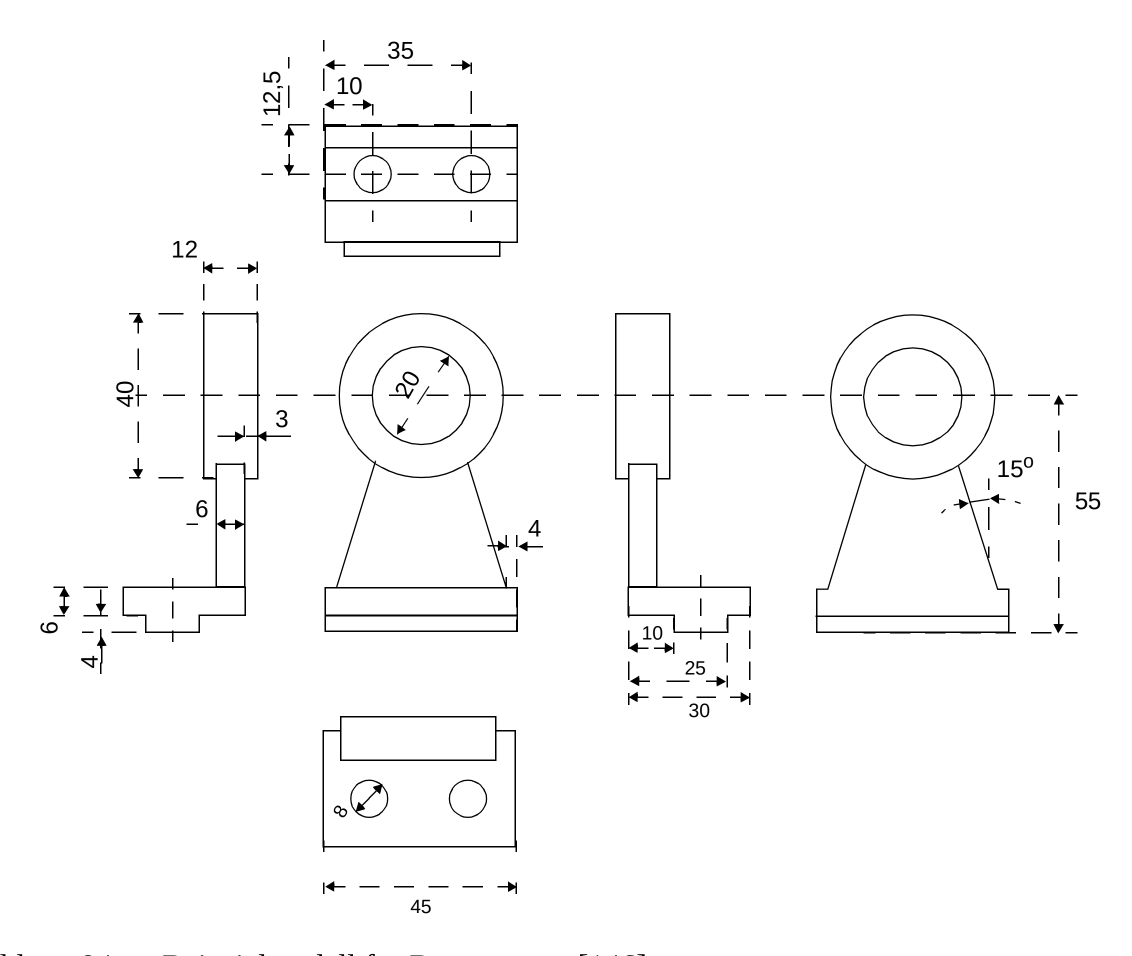


Abbildung 94: Beispielmodell für Benutzertest [146]

1. Auf kommerzielle Systeme wurde zurückgegriffen, da die in Kapitel 5.1.1.1 beschriebenen Systeme und Ansätze nicht sämtlich verfügbar waren bzw. keine hinreichend erfahrenen Benutzer zur Verfügung standen.

Die Aufgabe bestand darin, das in Abbildung 94 dargestellte Modell korrekt, d.h. unter Einhaltung der angegebenen Maße, und in möglichst kurzer Zeit zu modellieren. Das Modell wurde aus [146] entnommen und nachträglich von einem Maschinenbauingenieur bemaßt. Die Benutzer hatten Gelegenheit, die Aufgabe mehrfach zu üben, bis sich die Ausführungszeiten stabilisierten. Jeder Benutzer konnte die Vorgehensweise bei der Modellierung wählen, von der er glaubte, am schnellsten zum Ziel zu gelangen. Die schnellste Zeit wurde erfaßt und spiegelt sich in Abbildung 95 wider.

Die Tests wurden auf unterschiedlichen Rechnern durchgeführt, unterstellend daß die Rechnergeschwindigkeit bei diesem relativ einfachen Modell einen weitaus geringeren Einfluß auf die Konstruktionszeit hat, als die Bedienbarkeit des CAD-Systems. Da das Laufzeitverhalten verschiedener CAD-Systeme auch auf ein und derselben Hardware unterschiedlich ist, ist es praktisch unmöglich, den Einfluß der Rechenzeiten der Software auszuschalten.

<u>Ergebnisse</u>

Abbildung 95 stellt die Konstruktionszeiten, die mit den verschiedenen Systemen benötigt wurden, gegenüber. Mit ARCADE konnte je nach Vergleichssystem ein Geschwindigkeitsvorteil von einem Faktor 2 bis 4 erzielt werden.

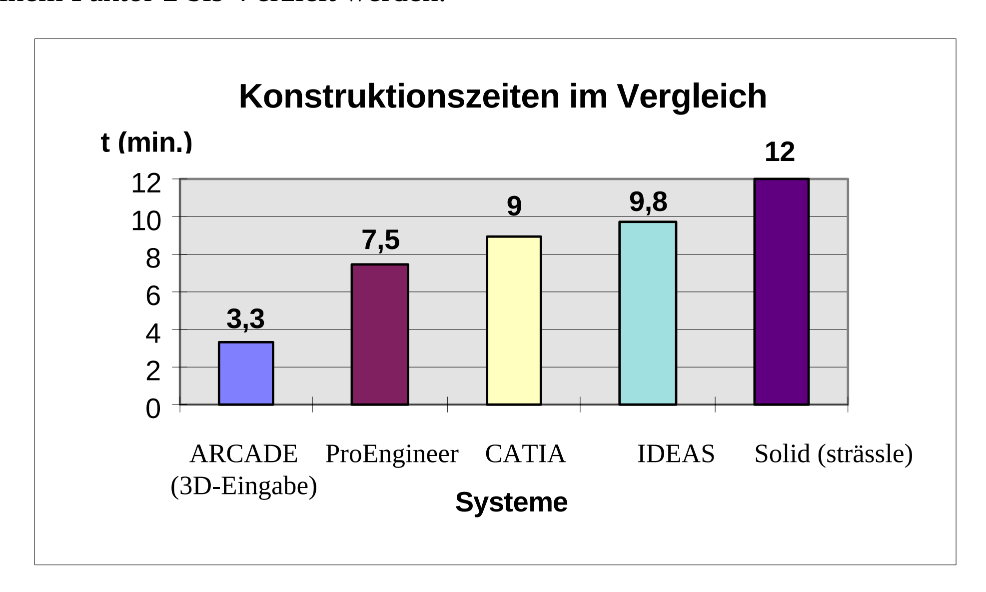


Abbildung 95: Konstruktionszeiten im Vergleich

Trotz der moderaten Komplexität des Bauteils kamen bei der Konstruktion mit ARCADE eine Vielzahl der neuartigen 3D-Interaktionstechniken zur Anwendung; im einzelnen waren dies:

- Objekterzeugung mittles diskretisierter 3D-Interaktionen.
- Diskretisierte Rotation und Translation (TCBM).
- Copy pick, history pick und history copy pick.
- Implizite Boolsche Operationen.

Die 3D-Interaktionen wurden durch das schnelle Verfahren zum Picken und Snappen unterstützt.

Bei der Versuchsdurchführung wurde ARCADE an einem herkömmlichen Arbeitsplatz mit konventionellem Monitor bedient. Die Ausgabe erfolgte monoskopisch. Als visuelle Form- und Tiefenhinweise waren die Wände und Schatten aktiv, das Modell wurde schattiert dargestellt.

Menüselektionen wurden mit der 2D-Maus durchgeführt, die graphischen Interaktionen mit der SpaceMouse, d.h. ein Umgreifen zwischen 2D- und 3D-Eingabegerät war nötig. Tastatureingaben wurden nicht getätigt. Der gesamte maßgetreue Modellierungsvorgang erfolgte graphisch-interaktiv. Im Gegensatz dazu kam keiner der Benutzer der kommerziellen CAD-Systeme ohne Tastatureingaben aus.

Der Geschwindigkeitsvorteil von ARCADE fällt umso bemerkenswerter aus, führt man sich die vergleichsweise hohen Anforderungen an die Hardware vor Augen, die die direkte Manipulation sowie der Einsatz des 3D-Cursors und der verwendeten Visualisierungstechniken stellt. Als Hardwareplattform für den Test mit ARCADE diente eine SGI Indigo2 mit einem High Impact Graphiksystem. Abbildung 96 zeigt das mit ARCADE erzeugte Beispielmodell.

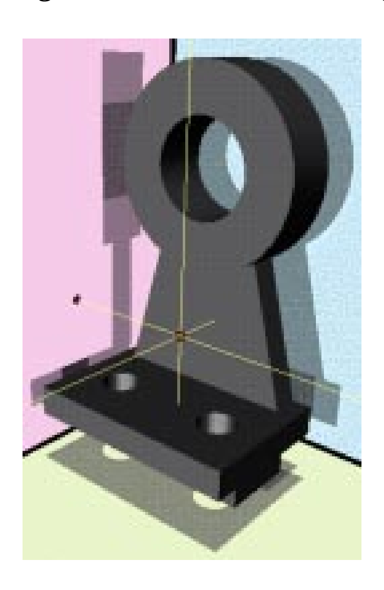


Abbildung 96: Ergebnismodell aus Benutzertest

5.1.2 Selektionsgeschwindigkeit mit verschiedenen Eingabemetaphern

Bei den im Verlauf der Arbeit durchgeführten Benutzertests zur Verbesserung der Interaktionstechniken ist aufgefallen, daß viele Testpersonen das am herkömmlichen Arbeitsplatz notwendige Umgreifen zwischen 3D- und 2D-Eingabegerät bemängelten. Um dieses Problem zu adressieren, wurde die 2D-Mauszeiger- und 3D-Cursor-Kontrolle in dem 3D-Einagegerät kombiniert. Ziel dabei war es:

- Die Fensteroberfläche auch mit dem 3D-Eingabegerät bedienen zu können.
- Die Selektion von Objekten und topologischen Elementen auch mit dem 2D-Cursor vornehmen zu können.
- Weiterhin die Freiheitsgrade des 3D-Eingabegerätes und die gesten-basierte Modifikationtechnik einsetzen zu können.

Bei dem kombinierten Ansatz wird mit dem 3D-Eingabegerät standardmäßig der 2D-Mauszeiger bewegt und kontext-abhängig der 3D-Cursor aktiviert. Die 2D-Maustasten können z.B. mit den Tasten der SpaceMaus simuliert werden.

Bewegt der Benutzer den 2D-Mauszeiger über das Graphikfenster, so wird mit jeder Bewegung ein Pickstrahl in die Szene geschickt. Trifft der Pickstrahl auf ein Objekt, wird an der entsprechenden Stelle die 3D-Pickkugel eingeblendet und mit dem beschriebenen schnellen Picking- und Snapping-Verfahren kann eine präzise Selektion von Objekten und ihren topologischen Elementen

erfolgen. Sobald ein Objekt oder topologisches Element gepickt wurde, greifen die beschriebenen Interaktionstechniken für 3D-Eingabegeräte, wie z.B. die TCBM.

Befindet sich der Benutzer in der Objekterzeugung, kann an der gepickten Position ein Objekt mit dem 3D-Eingabegerät kreiert werden. Ist der Benutzer in der Objekterzeugung und der 2D-Mauszeiger befindet sich nicht über einem in der Szene existierenden Objekt, so wird mit dem ersten Pick des Benutzers der 3D-Cursor aktiviert und er kann in gewohnter Weise ein Objekt im freien Raum erzeugen.

Am Monitor kann die vorgestellte Kombination aus 2D- und 3D-Interaktion mit einem 3D-Eingabegerät die Akzeptanz seitens unerfahrener Benutzer steigern, da das Umgreifen entfällt und die 3D-Interaktionstechniken für schnelles und präzisen Modellieren mit 3D-Eingabegeräten weiterhin genutzt werden können. Erfahrene Benutzer können 2D- und 3D-Maus zur 2-händigen Eingabe nutzen und z.B. Menüselektionen mit der nicht-dominanten Hand durchführen während die dominante Hand an der 3D-Maus bleibt, so daß sich dieser Modus für sie eher negativ auswirkt.

Der im folgenden beschriebene Test geht der Frage nach, wie sich die Selektionsgeschwindigkeit mit 2D-Maus und 2D-Zeiger gegenüber der mit 3D-Eingabegerät und 3D-Cursor verhält und inwieweit die Selektionsgeschwindigkeit durch das Steuern des 2D-Zeigers mit dem 3D-Eingabegerät positiv beeinflußt werden kann.

<u>Testdurchführung</u>

Die Aufgabe bestand darin, in der in Abbildung 97 dargestellten Szene mit 3 Objekten sukzessive eine Fläche, Kante bzw. Ecke je eines Objektes zu selektieren und zu deselektieren. Das graphische Echo mußte zunächst aus der Bildschirmmitte zu Objekt A, dann zu Objekt B und schließlich zu Objekt C bewegt werden. Die Objekte lagen in unterschiedlichen Tiefenebenen, wie sich aus den Schatten erkennen läßt. Dieser Vorgang sollte mit drei verschiedenen Verfahren möglichst schnell durchgeführt werden. Folgende Verfahren wurden evaluiert:

1) 3D-Maus kontrolliert 3D-Zeiger, wobei zur Selektion das in dieser Arbeit entwickelte Pickingverfahren eingesetzt wurde.
2a) 3D-Maus kontrolliert 2D-Zeiger gemäß 2D-Maus, d.h. Bewegungen der 3D-Maus vom Benutzer weg, führen zu Aufwärtsbewegungen des 2D-Zeigers (keine Reiz-Reaktions-Korrespondenz!).
2b) 3D-Maus kontrolliert 2D-Zeiger gemäß 3D, d.h. Bewegungen der 3D-Maus nach oben führen zu Aufwärtsbewegungen des 2D-Zeigers (Reiz-Reaktions-Korrespondenz!).
3) 2D-Maus kontrolliert 2D-Zeiger.

Die Selektion wurde bei Verfahren 2a, 2b und 3 mit einem konventionellen Pickstrahl durchgeführt.

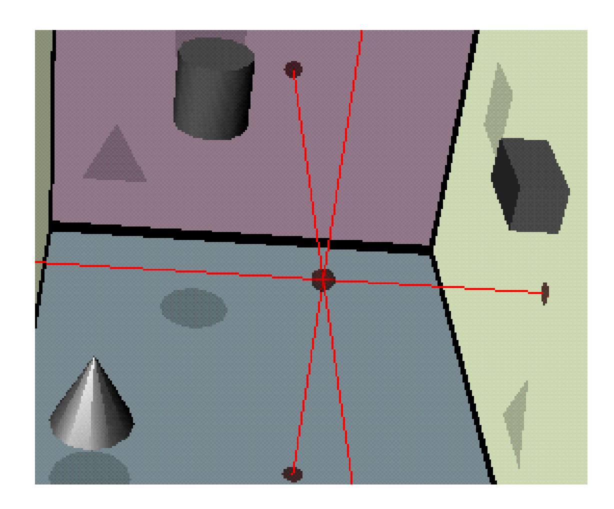


Abbildung 97: Szene zur Evaluierung der Selektionsgeschwindigkeit

Der Test wurde mit je 6 Personen durchgeführt, die im Umgang mit 3D-Eingabegeräten erfahren bzw. unerfahren waren; alle Personen waren im Umgang mit der 2D-Maus erfahren. Die Testpersonen hatten Gelegenheit, sich mit den einzelnen Verfahren vertraut zu machen, bis sich die Ausführungszeiten stabilisierten. Dann wurden zwei Versuchsreihen durchgeführt, wobei in jeder Versuchsreihe das Verfahren 1 und 3 sowie das Verfahren 2a oder 2b eingesetzt wurde. Die Reihenfolge der Verfahren wurde für die verschiedenen Testpersonen variiert, um Auswirkungen von Lerneffekten auf die Testergebnisse zu minimieren. Für die erfahrenen Testpersonen war Verfahren 2b Bestandteil der ersten Testreihe und Verfahren 2a Teil der zweiten Testreihe; bei den unerfahrenen Benutzern wurde umgekehrt vorgegangen. Jeder Benutzer führte jedes Verfahren in beiden Versuchsreihen viermal hintereinander aus. Währenddessen wurde die Zeit für die Selektionsaufgabe gemessen.

Der Test wurde an einem ’herkömmlichen’ Arbeitsplatz mit Monitor im monoskopischen Betrieb durchgeführt. Als 3D-Eingabegerät wurde eine SpaceMouse verwendet. Die im folgenden dargestellten Ergebnisse gelten somit für eine Umgebung, in der ’indirekt’ interagiert wird, d.h. die wahrgenommene Position des graphischen Echos entspricht nicht der Position der Hand des Benutzers im Raum.

Ergebnisse

Abbildung 98 zeigt die durchschnittliche Ausführungszeit der Selektionsaufgabe für die beiden Personengruppen. Zu erkennen ist, daß die im Umgang mit 3D-Eingabegeräten erfahrenen Benutzer im Mittel mit allen drei Verfahren schneller waren als die unerfahrenen. Besonders groß ist die Diskrepanz bei der Verwendung von 3D-Eingabegerät und 3D-Cursor; hier sind erfahrene Benutzer eineinhalb mal so schnell wie unerfahrene. Bei Verwendung einer 2D-Maus fällt der Unterschied zwischen den Gruppen kaum mehr ins Gewicht. Interessant ist hierbei insbesondere der Vergleich mit dem Verfahren 1 (3D-Eingabegerät und 3D-Cursor): erfahrene Benutzer sind bei Verwendung der 2D-Maus ca. dreimal, unerfahrene Benutzer sogar viermal so schnell. Verfahren 2 - die Kombination aus 3D-Eingabegerät und 2D-Zeiger - stellt für beiden Gruppen einen Kompromiß dar, wobei dieser für die erfahrenen Benutzer ungünstiger ausfällt als für die unerfahrenen.

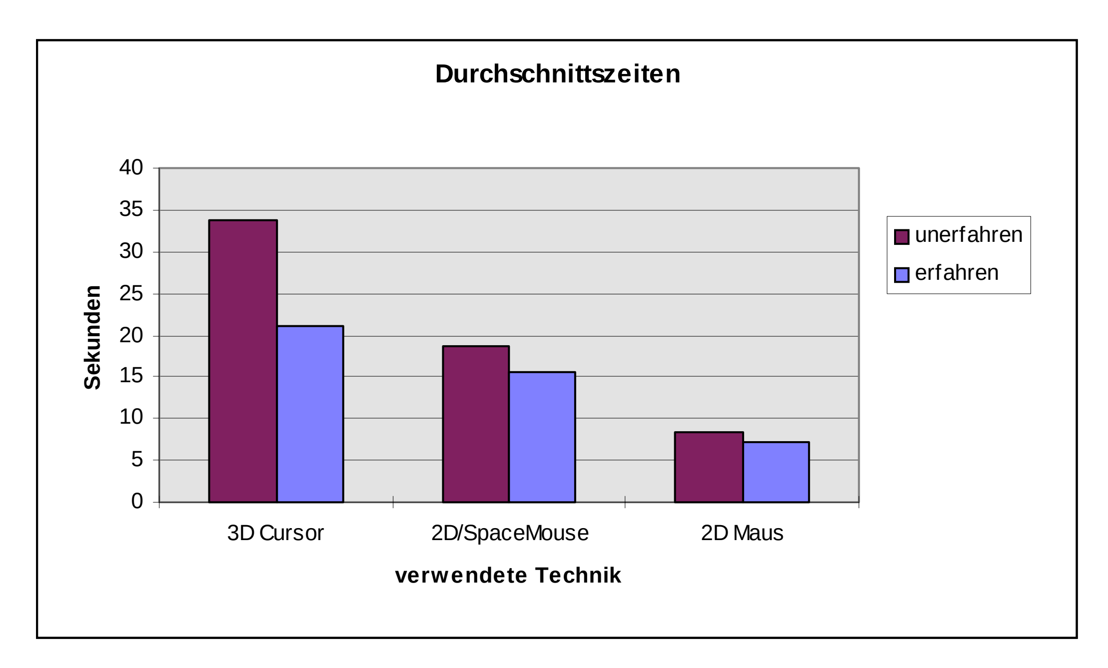


Abbildung 98: Durchschnittliche Selektionszeit der drei Verfahren

Ein detaillierterer Blick in die Versuchsreihen bestätigt die Hypothese, daß die Verfahren 2a und 2b von den beiden Gruppen unterschiedlich angenommen wurden. Die mit 3D-Eingabe erfahrenen Testpersonen, die mit Reiz-Reaktions-Korrespondenz in 3D vertraut sind, empfanden Verfahren 2b als intuitiver und erzielten im Vergleich zu Verfahren 2a die besseren Zeiten. Bei den unerfahrenen Testpersonen verhielt es sich genau umgekehrt. Daraus läßt sich folgern, daß die mit 3D-Eingabe unerfahrenen Benutzer durch den täglichen Gebrauch der 2D-Maus die natürliche und in realen Umgebungen vorherrschende Reiz-Reaktions-Korrespondenz in 3D als ungewohnt und weniger intuitiv empfinden. Unter dieser Beobachtung leiden viele aus der Literatur bekannte Systemvergleiche virtueller mit herkömmlichen Umgebungen, z.B. [166]. Durch den täglichen Gebrauch der Maus sind viele Computerbenutzer und somit potentielle Testpersonen in einer Art und Weise vorgeprägt, die einen objektiven Vergleich behindern. Die von unbedarften Testpersonen oft geäußerte Begeisterung gegenüber virtuellen Umgebungen geht nicht notwendigerweise mit einer effizienteren Systemnutzung einher. Daher sind Vergleiche der *expert performance* i.d.R. aufschlußreicher als Evaluierungen mit Laien, wenn es um echte 3D-Interaktion in virtuellen Umgebungen geht.

Abbildung 99 zeigt den Verlauf der Versuche mit den beiden Varianten von Verfahren 2 bezüglich der beiden Testgruppen. In der ersten Hälfte der Testreihe (Durchgang 1 bis 4) arbeitete die Gruppe der erfahrenen Benutzer mit Verfahren 2b (3D-Eingabegerät kontrolliert 2D-Zeiger bei Wahrung der Reiz-Reaktions-Korrespondenz), die Gruppe der unerfahrenen Benutzer arbeitete mit Verfahren 2a (3D-Eingabegerät kontrolliert 2D-Zeiger gemäß 2D-Maus, d.h. ohne Wahrung der Reiz-Reaktions-Korrespondenz). In der zweiten Hälfte der Versuchsreihe arbeiteten die Testpersonen mit dem jeweils anderen Verfahren. Aus Abbildung 99 ist sehr gut zu erkennen, daß die jeweils vertrautere Metapher bei der entsprechenden Benutzergruppe auch zu schnelleren Ausführungszeiten führte (je Benutzergruppe liegen die Zeiten in der linken Hälfte im Durchschnitt unter den Zeiten in der rechten Hälfte). Interessant sind die Auswirkungen beim Wechsel des Verfahrens: die Gruppe der erfahrenen Benutzer wird langsamer als die unerfahrene Gruppe, kann aber im Verlauf der weiteren Schritte wieder Zeit gutmachen. Die drastische Verschlechterung der erfahrenen Gruppe ist auf eine Testperson zurückzuführen, die mit Verfahren 2a große Schwierigkeiten hatte. Betrachtet man das Ergebnis dieser Testperson als Ausreißer, fällt das

Gruppenergebnis besser aus, die Tendenz und eine deutliche Verschlechterung für Durchgang 5 bleibt allerdings erhalten. Die Zeitmessungen decken sich mit den subjektiven Einschätzungen der Testpersonen. Die unerfahrene Gruppe hat Verfahren 2a, die erfahrene Gruppe Verfahren 2b als intuitiver eingestuft.

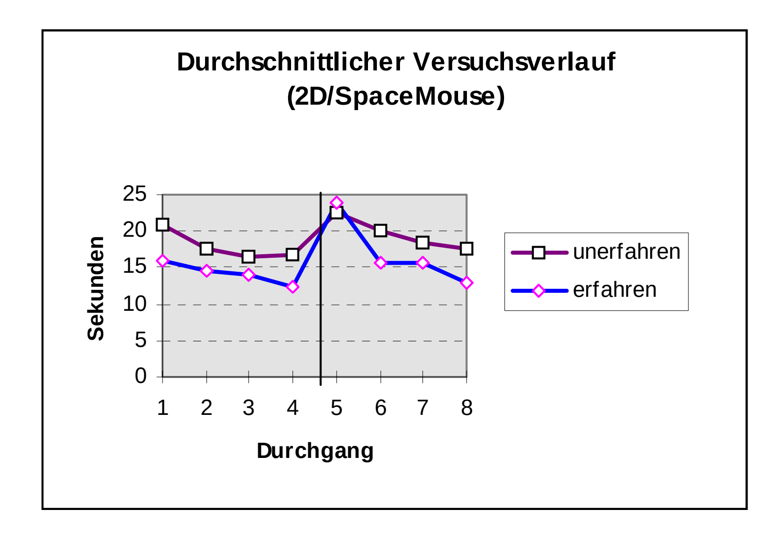


Abbildung 99: Versuchsverlauf mit Verfahren 2

Zusammenfassend kann festgestellt werden, daß die Kombination aus 3D-Eingabegerät und 2D-Zeiger bei der Selektionsaufgabe gerade unerfahrenen Benutzern Vorteile bietet. Mit 2D-Maus und 2D-Zeiger lassen sich bei dieser Aufgabe zwar die besten Ausführungszeiten erzielen, allerdings werden dadurch die Vorteile echter 3D-Interaktion aufgegeben. Die kontext-sensitive Steuerung des 2D- und 3D-Zeigers mit einem 3D-Eingabegerät ebnet somit den Übergang von bekannten Systemen mit 2D-Eingabe hin zu Systemen mit 3D-Eingabe insbesondere für Benutzer, die damit (noch) nicht vertraut sind.

5.1.3 Evaluierung der Methode zur Zusammenbauunterstützung

In Kapitel 4.2.11 wurde die Behauptung aufgestellt, daß ohne Systemunterstützung virtuelle Zusammenbauprozesse kaum interaktiv durchzuführen sind, da der Benutzer eine Baugruppe nur durch die visuelle Kontrolle nicht genau positionieren und orientieren kann. Um diesem Problem zu begegnen, sind Ansätze mit Kollisionserkennung und den damit verbundenen bereits erläuterten Vor- und Nachteilen weit verbreitet. In diesem Abschnitt wird die entwickelte Methode zur Zusammenbauunterstützung dem manuellen Verfahren gegenübergestellt, bei dem nur die Visualisierung der Objekte sowie ihrer Schatten Aufschlüsse über die richtige Position liefern.

Testdurchführung

Als Aufgabe galt es, einen Bolzen mit Hilfe der SpaceMouse durch graphische Interaktion mit sechs Freiheitsgraden in der dafür vorgesehenen Lagerung eines Bolzenhalters zu plazieren. Abbildung 100 zeigt die Ausgangssituation des Versuchsaufbaus. Der Bolzen muß innerhalb des Bolzenhalters richtig positioniert und orientiert werden und darf diesen nicht durchdringen.

Gemessen wurde beim manuellen Verfahren die Zeit, bis der Benutzer der Meinung war, er könne den Bolzen nicht korrekter plazieren sowie die Abweichung bezüglich Rotation und Translation von der Zielposition; die Rotation um die Längsachse des Bolzens spielte dabei keine Rolle. Beim assistierenden Verfahren wurde die Zeit gemessen, die jede Testperson brauchte, um den Bolzen in

die Endlage zu bewegen. Der Test wurde von fünf mit der SpaceMouse vertrauten Personen durchgeführt.

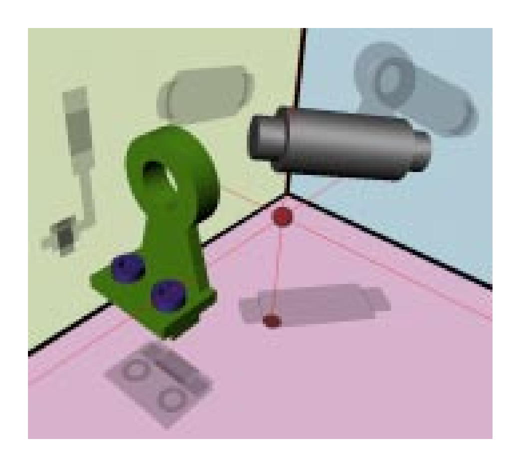
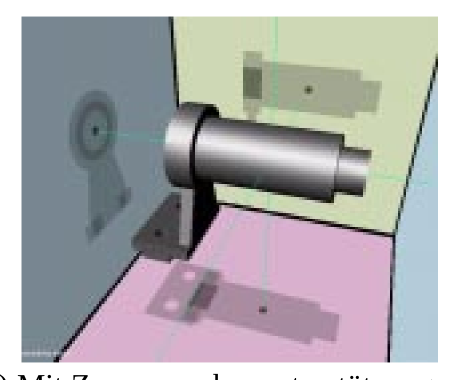


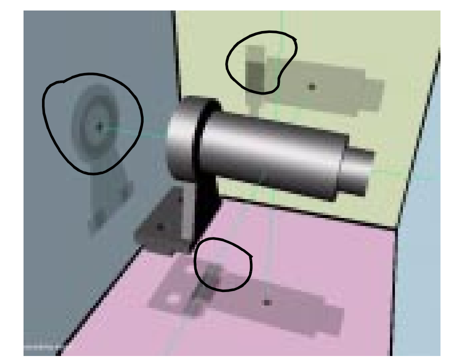
Abbildung 100: Ausgangssituation für den Zusammenbauversuch

Ergebnisse

Die Testpersonen nahmen sich im Durchschnitt 48 Sekunden Zeit, um den Bolzen zu positionieren, wenn sie auf sich gestellt waren. Wurden sie vom System unterstützt, so konnte der Bolzen im Durchschnitt nach 5,5 Sekunden in die gewünschte Position gebracht werden. Mehr noch als diese Zeitdifferenz von fast einem Faktor 10, stützen die Reaktionen der Benutzer während der Testdurchführung die oben aufgestellte Behauptung. Arbeiteten die Testpersonen ohne Systemunterstützung, so waren sie mit ihrer Leistung mehr oder weniger unzufrieden. Trotz offensichtlicher Abweichungen des Bolzens von der gewünschten Position bzw. Durchdringungen mit dem Bolzenhalter, gaben die Benutzer mit der Aussagen auf, den Bolzen interaktiv nicht besser positionieren zu können. Abbildung 101 zeigt links den mit Hilfe der Zusammenbauunterstützung eingebauten Bolzen und rechts eine typische Situation nach dem Versuch, den Bolzen ohne Systemunterstützung richtig zu positionieren.



a) Mit Zusammenbauunterstützung



b) Ohne Zusammenbauunterstützung

Abbildung 101: Typische Situation nach dem Zusammenbauversuch mit und ohne Systemunterstützung

Zusammenfassend läßt sich feststellen, daß dieser Versuch zeigt, daß interaktiver, virtueller Zusammenbau eine Systemunterstützung benötigt. Das vorgestellte Assembly Assistance Verfahren ist in der Lage, den Zusammenbau auf effiziente Art zu unterstützen.

5.2 Evaluierung der visuellen Tiefenhinweise

Im folgenden wird der Einfluß der Form- und Tiefenhinweise auf die Wahrnehmung der Szene am Bildschirm und deren Einfluß auf die Interaktion untersucht. Im Mittelpunkt der Betrachtungen stehen der Stereoeffekt und der Schattenwurf. Um herauszufinden, wie stark die einzelnen visuellen Hilfsmechanismen den Benutzer bei der Wahrnehmung und 3D-Interaktion unterstützen, wurden vier Experimente durchgeführt, die im folgenden vorgestellt werden:

- Erkennen von relativer Tiefe bei stereoskopischer Darstellung

  Dieser Versuch soll klären, ob eine stereoskopische Darstellung auf einem herkömmlichen Monitor räumliche Tiefenhinweise liefert.
- Wechselwirkung zwischen Tiefe und Größe

  Dieser Versuch geht der Frage nach, ob die Tiefeninformation einer stereoskopischen Darstellung die durch die Größe der Objekte suggerierte Entfernung überdeckt.
- Wechselwirkung zwischen Tiefe und Überdeckung

  Dieses Experiment untersucht, ob die Tiefeninformation oder die Verdeckung die für den Betrachter wichtigere Information bei der Entfernungsabschätzung darstellt.
- Einfluß verschiedener visueller Tiefenhinweise auf die Navigation mit dem 3D-Cursor im Raum

  Mit dieser Versuchsreihe soll bestimmt werden, welche der unterstützenden Visualisierungstechniken, die Navigation des 3D-Cursors durch den Raum wie beeinflußt.

5.2.1 Erkennen von relativer Tiefe bei stereoskopischer Darstellung

<u>Versuchsaufbau</u>

Zwei gleich große Objekte (Würfel) sind in unterschiedlichen Tiefenebenen im Raum positioniert. Beide schneiden sich nicht und werden parallel projiziert, so daß sie auf dem Bildschirm gleich groß erscheinen (siehe Abbildung 102). Links ist die von den Benutzern wahrgenommene Parallelprojektion der Szene auf die xy-Ebene dargestelltt, rechts ist die Lage der beiden Objekte in verschiedenen Tiefenebenen von oben betrachtet nochmals verdeutlicht.

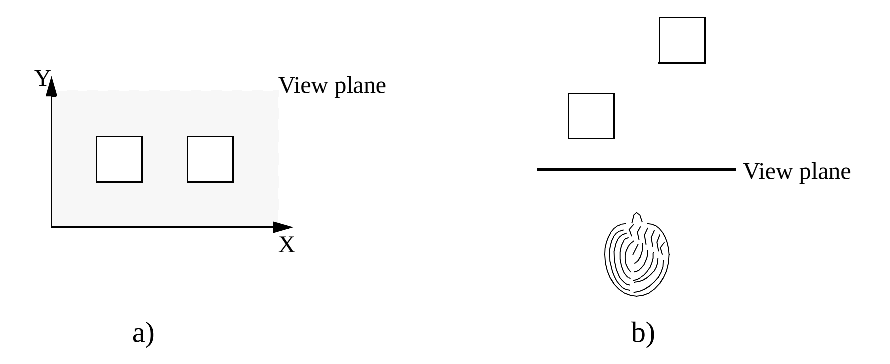


Abbildung 102: Szene aus dem 1. Experiment: von vorne (a); von oben (b)

<u>Versuchsdurchführung</u>

Acht Testpersonen wurde obige Szene einmal monoskopisch und einmal stereoskopisch dargestellt - außer dem Stereoeffekt waren keine Tiefenhinweise in der Szene enthalten. Die Betrachter

sollten das weiter vorne liegende Objekt identifizieren. Zur Stereodarstellung wurden Shutter-Brillen benutzt.

Testergebnis

Sieben von acht Testpersonen konnten den weiter vorne liegenden Würfel im Stereomodus eindeutig identifizieren. Während des Tests fiel auf, daß die Betrachter unterschiedlich lange brauchten, um den Stereoeffekt wahrzunehmen. Eine Testperson benötigte mehr als fünf Minuten, bis sich bei ihr das Gefühl von Tiefenwahrnehmung einstellte; allerdings klassifizierte sie die Würfel trotzdem falsch. Das Phänomen der 3D-Blindheit ist aus der Literatur bekannt [123]; ob es sich um eine 3D-blinde Person handelte, kann bezweifelt werden, da sich die 3D-Wahrnehmung mit fortschreitender Zeit verbesserte und die Person in der realen Welt keine Probleme mit der Tiefenwahrnehmung hat. Offenbar gibt es Menschen, die sehr lange Adaptionszeiten an das 3D-Sehen mit Shutter-Brillen benötigen.

5.2.2 Wechselwirkung zwischen Tiefe und Größe

Versuchsaufbau

Die Szene enthielt wieder zwei unterschiedlich weit vom Betrachter entfernte Objekte. Allerdings ist diesmal das nähere Objekt kleiner und das weiter entfernte größer (siehe Abbildung 103). Damit wird der Effekt der scheinbaren Größe als Tiefenhinweis ad absurdum geführt.

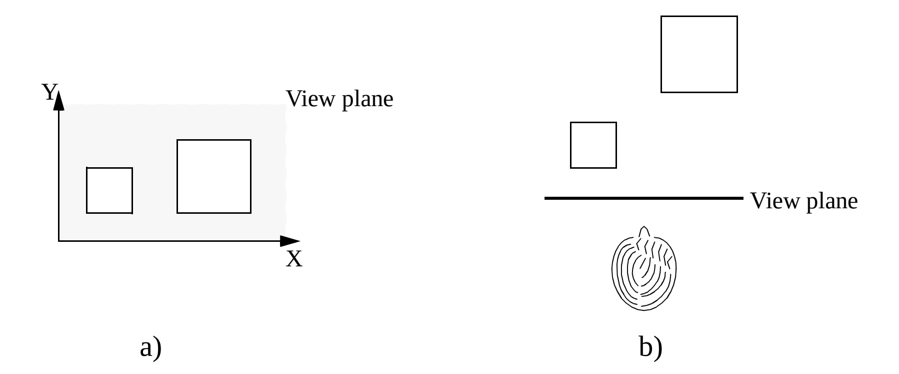


Abbildung 103: Stereo vs. ’scheinbare Größe’: von vorne (a) und von oben (b)

Testdurchführung

Die Szene wurde den Testpersonen stereoskopisch präsentiert. Die Testpersonen waren erneut gefragt, das ihnen näher liegende Objekt zu identifizieren.

Testergebnis

Trotz der unterschiedlichen Objektgröße, die nahelegt, daß das größere Objekt näher ist, waren alle Testkandidaten in der Lage, die relative Lage der Objekte im Stereobetrieb richtig einzuschätzen. Offensichtlich überwiegt der Stereoeffekt den Effekt der ’scheinbaren Größe’, was als positiv beurteilt werden kann, da CAD-Modelle oft in verschiedenen Abmessungen vorliegen und davon auszugehen ist, daß sich die Tiefenwahrnehmung dadurch nicht oder nur wenig beeinflussen läßt.

Der Versuch wurde mit drei Objekten von unterschiedlicher Größe wiederholt. Erstaunlicher Weise empfanden die Testkandidaten die Erkennung der Tiefenlage als einfacher, obwohl mehr Objekte in der Szene waren. Offensichtlich benutzt das menschliche Gehirn die einzelnen Objekte als Bezugspunkte für die Erkennung der Tiefe und somit für das räumliche Sehen.

5.2.3 Wechselwirkung zwischen Tiefe und Überdeckung

Versuchsaufbau

Ein Würfel und ein L-förmiger Sweepkörper wurden so positioniert, daß bei dem Betrachter der Eindruck einer Überdeckung entstand (siehe Abbildung 104a). Der L-förmige Sweepkörper ist dem Betracher näher als der Würfel (siehe Abbildung 104b).

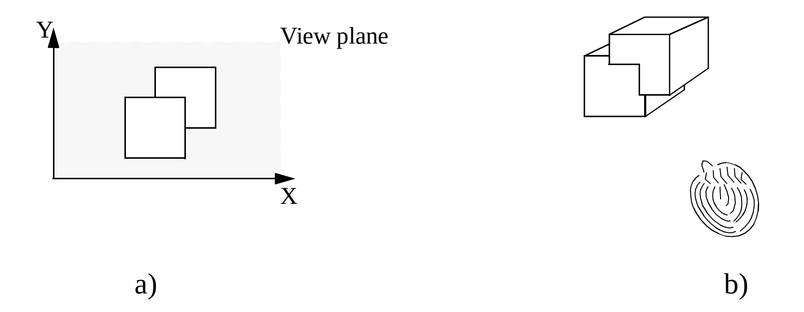


Abbildung 104: Szene aus dem 3. Experiment: von vorne (a) und von oben (b)

Testdurchführung

Die Szene wurde den Testpersonen stereoskopisch präsentiert. Wieder waren die Testpersonen gefragt, das ihnen näher liegende Objekt zu identifizieren.

Testergebnis

Nur 50% der Testpersonen haben das scheinbar verdeckte Objekt als vorderes identifiziert, d.h. Verdeckung ist für das menschliche Wahrnehmungssystem so bedeutsam, daß sie die Disparität der Bilder dominiert.

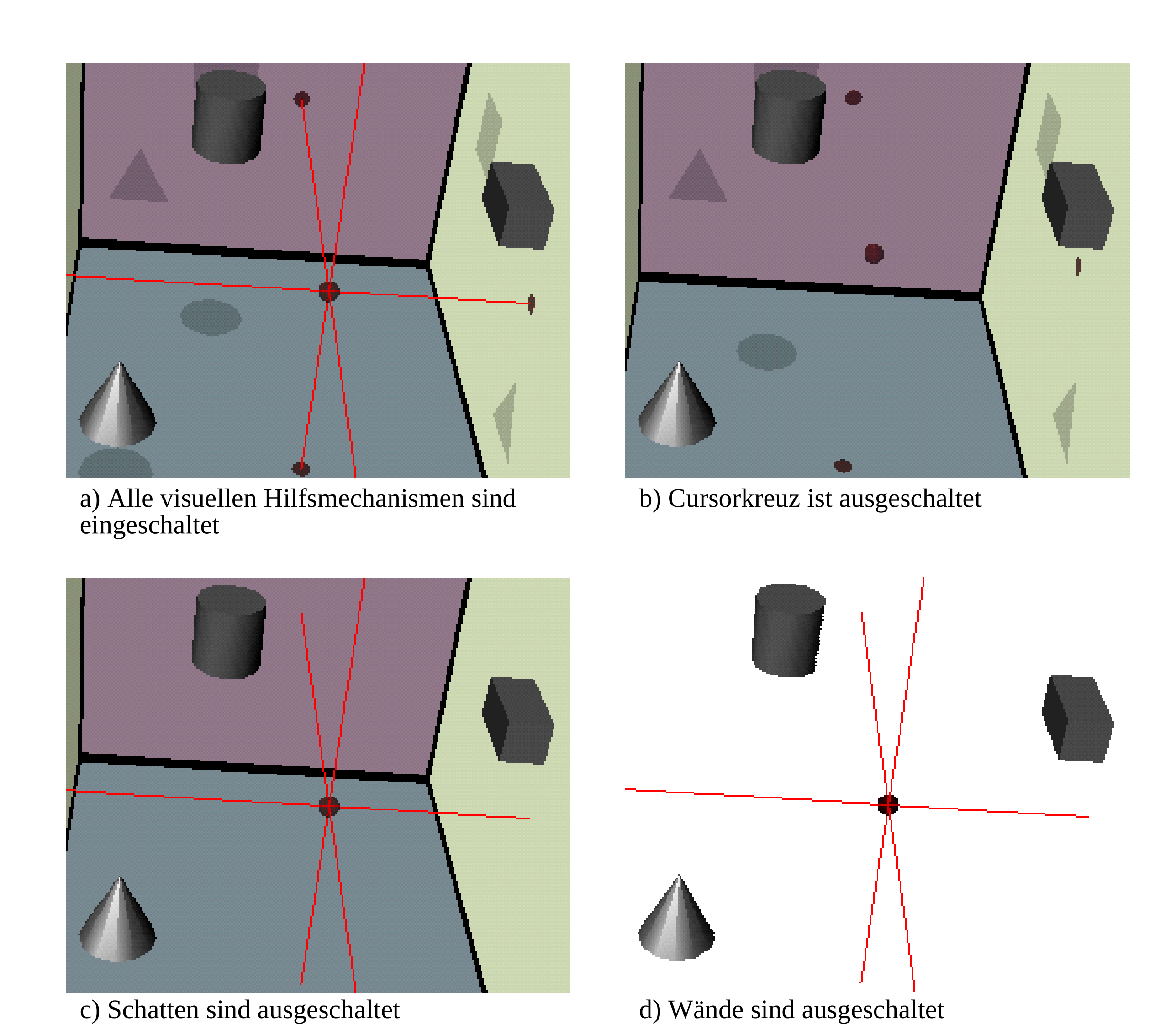
An dieser Stelle sei nochmals angemerkt, daß der von der Stereoprojektion hervorgerufene Effekt nicht mit realistischen Szenen vergleichbar ist, da Konvergenzeffekte fehlen und Akkomodation unanhängig von der scheinbaren Entfernung der virtuellen Objekte stets auf den Bildschrim erfolgt.

5.2.4 Einfluß visueller Tiefenhinweise auf die Navigation mit dem 3D-Cursor

In diesem Test ging es darum, die verschiedenen Visualisierungshilfen, die in ARCADE implementiert wurden, auf ihren Einfluß auf die Interaktion mit dem 3D-Cursor im Raum zu untersuchen. Betrachtet wurden das Cursorkreuz, die Wände, der Schattenwurf von Objekten sowie der Stereoeffekt.

Testaufbau

Abbildung 105 zeigt die verwendete Szene mit unterschiedlichen Kombinationen von visuellen Tiefenhinweisen. Die Aufgabe für die Testpersonen bestand darin, den 3D-Cursor mit der Space-Mouse von dem Mittelpunkt der Szene aus von einem Objekt zum nächsten zu bewegen und jeweils ein topologisches Element zu picken. Dabei sollte zunächst eine Fläche eines Objektes, dann eine Kante eines anderen und schließlich eine Ecke des dritten Objektes gepickt werden. Welches topologische Element eines Objektes gepickt wird und in welcher Reihenfolge die einzelnen Objekte angesteuert werden, war den Testkandidaten überlassen; sie sollten dies allerdings möglichst schnell tun.



a) Alle visuellen Hilfsmechanismen sind eingeschaltet

b) Cursorkreuz ist ausgeschaltet

c) Schatten sind ausgeschaltet

d) Wände sind ausgeschaltet

Abbildung 105: Beispielszenario zur Evaluierung der visuellen Hinweise

Testdurchführung

Die Testreihe wurde mit 8 Personen durchgeführt. Die Gruppe von Testpersonen wurde in bezüglich des Umgangs mit der SpaceMouse erfahrene und unerfahrene Personen aufgeteilt. Jede Teilgruppe enthielt 4 Personen. Die Testreihe lief mit jeder Person in zweimal 5 Schritten ab, wobei die Visualisierungshilfen gemäß Tabelle 11 aktiviert wurden. Um eine Abhängigkeit der Ergebnisse von der Reihenfolge der Testschritte zu vermeiden, führte je eine Hälfte einer Gruppe

die Tests in umgekehrter Reihenfolge durch. Die fünf Testschritte wurden zweimal mit und zweimal ohne stereoskopische Darstellung durchgeführt. Dabei wurde die von den Kandidaten benötigte Zeit erfaßt und das schlechtere Ergebnis gestrichen. Die unerfahrenen Benutzer hatten Gelegenheit, sich mit der prinzipiellen Bedienung der SpaceMouse vertraut zu machen.

Das Maß des Stereo-Effekte, d.h. den Abstand der virtuellen Kameras, konnten die Benutzer individuell einstellen und dieses somit an ihre Wahrnehmung anpassen Die durch das Stereo-Rendering bedingte Verzögerung der Darstellungsgeschwindigkeit wurde durch Anpassen der Translationssensitivität des 3D-Cursors kompensiert.

Tabelle 11: Kombinationen von visuellen Tiefenhinweisen für den Benutzetest

WÄNDE	SCHATTEN	CURSORKREUZ
-	-	-
-	-	X
X	-	X
X	X	-
X	X	X

Testergebnis

Aus Abbildung 106 geht hervor, daß die stereoskopische Darstellung dann Vorteile bringt, wenn die Schattenprojektion deaktiviert ist. Dies deckt sich mit dem subjektiven Eindruck vieler Testpersonen, die die Tiefenwirkung des Stereoeffektes und den durch die Schatten hervorgehobenen Tiefeneindruck als widersprüchlich empfanden.

Die beste Zeit wurde im monoskopischen Modus mit der Kombination von Wänden und Schatten erzielt. Die zweitbeste Zeit wurde im Stereomodus in Kombination mit Wänden und Cursorkreuz erzielt. Letztere wurde unter den Kombinationen mit Stereo-Ausgabe auch subjektiv von den Testpersonen bevorzugt.

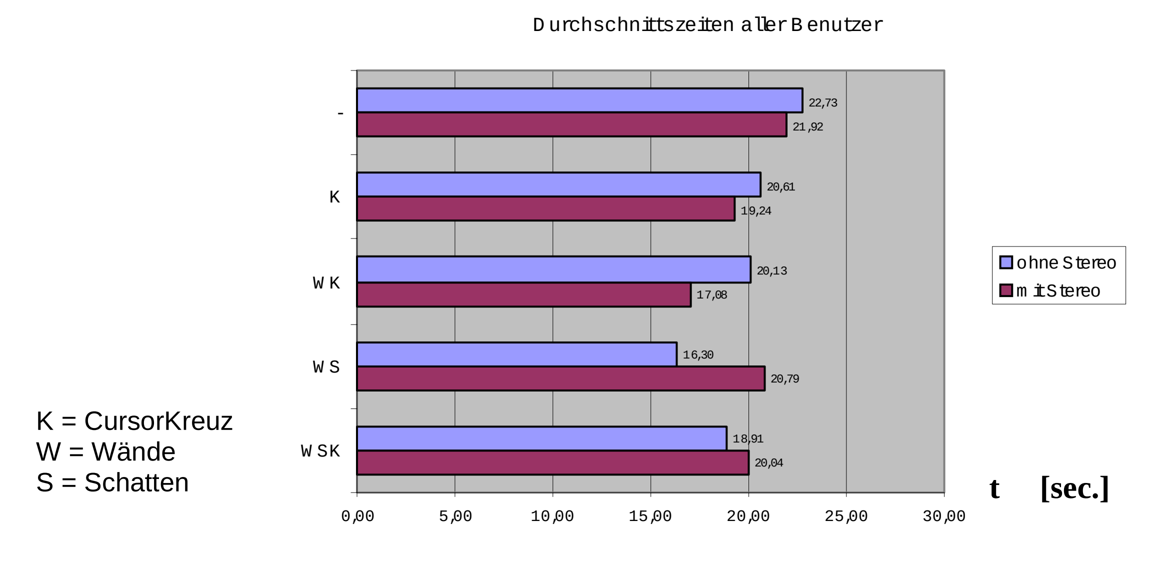


Abbildung 106: Einfluß der Visualisierung auf die Interaktion

Die bisherige Auswertung gestattet noch keine Aussage darüber, wie groß der Einfluß einzelner visueller Hinweise auf die Aufgabenbearbeitung war. Betrachtet man die Gruppe der erfahrenen SpaceMouse-Benutzer, so kommt man zu einem erstaunlichen Ergebnis: Die visuellen Hinweise

bringen im monoskopischen Modus keine signifikanten Vorteile. Für den Stereo-Modus gilt das bislang gesagte.

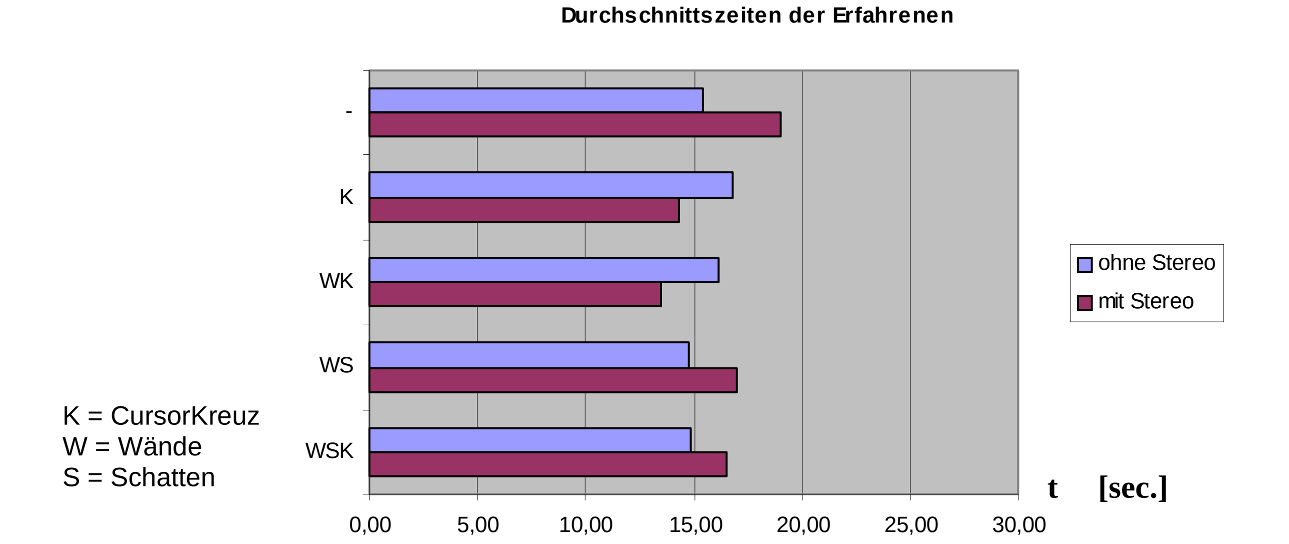


Abbildung 107: Einfluß der Visualisierung auf die Interaktion erfahrener Benutzer

Dieser Effekt läßt sich durch eine Beobachtung erklären, die während der Testdurchführung selbst bei erfahrenen Kandidaten gemacht wurde: Die Cursorpositionierung wurde zunächst grob in der Bildschirmebene durchgeführt, dann wurde der 3D-Cursor in der Tiefe verschoben, bis das gewünschte Element geschnitten wurde. Durch die Transparenz der Pickkugel ist man mit dieser Strategie weitgehend unanhängig von den visuellen Tiefenhinweisen. Nur Benutzer, die ARCADE kannten, haben mit dem 3D-Cursor Diagonalbewegungen im Raum durchgeführt und kamen damit absolut betrachtet zu den besten Ergebnissen.

Bei den unerfahrenen Benutzern macht sich bei der Stereodarstellung wieder der schon bekannte Effekt bemerkbar, daß Stereo und Schatten sich gegenseitig negativ beeinflussen (vgl. Abbildung 108). Diesmal ist als Tendenz zu erkennen, daß die visuellen Hinweise die Bearbeitungsgeschwindigkeit günstig beeinflussen. Es läßt sich allerdings nicht feststellen, daß bei der gewählten Aufgabe ein bestimmter visueller Hinweis generell positiv wirkt. Es läßt sich auch nicht erklären, warum das Cursorkreuz im letzten Fall dazu führt, daß sich die Aufgabenbearbeitung verlangsamt.

Durchschnittszeiten der Unerfahrenen

Abbildung 108: Einfluß der Visualisierung auf die Interaktion unerfahrener Benutzer

Wenn es aus den Ergebnissen der Testreihe auch nicht gelingt, das Maß des Einflusses einzelner visueller Hinweise zu extrahieren, so läßt sich jedoch festhalten, daß mit den visuellen Tiefenhinweisen stets eine schnellere Aufgabenbearbeitung möglich war, als ohne solche Hinweise nur mit stereoskopischer Darstellung.

Daraus läßt sich für das Arbeiten am herkömmlichen Monitor folgern, daß eine Stereoausgabe nicht zwingend erforderlich ist. Die indirekte 3D-Interaktion mit einem tischgebundenen 3D-Eingabegerät ist auch ohne 3D-Ausgabe sinnvoll möglich. Im täglichen Umgang mit dem System ARCADE wird der Wert der visuellen Tiefenhinweise - in erster Linie der Schatten - augenfällig: Gerade bei der Objekterzeugung und -positionierung bieten die Schatten wertvolle Hinweise über die räumlichen Relationen der Objekte in der Szene. Die Transparenz der projizierten Schatten ermöglicht auf einen Blick das Erkennen von Objektdurchdringungen bzw. -überlappungen.

Anders sieht die Situation am Virtuellen Tisch aus; hier wird Stereoprojektion zu einer unentbehrlichen Voraussetzung, möchte man in dem Raum über dem Tisch direkt interagieren. Von der hier vorgestellten Schattenprojektion sollten dann aus den genannten Gründen abgesehen werden. Interessant wären Untersuchungen im Rahmen weiterführender Aktivitäten, inwiefern ein dynamischer Schattenwurf zwischen Objekten in Kombination mit stereoskopischer Ausgabe sinnvoll ist, ob er Vorteile bringt und wie diese dem Berechnungsaufwand gegenüberstehen.

5.3 Zusammenfassung

In diesem Kapitel wurden die im Rahmen der vorliegenden Arbeit entwickelten 3D-Interaktions- und Visualisierungstechniken evaluiert. Die Interaktionstechniken wurden mit bekannten Ansätzen aus der Literatur auf Basis theoretischer Überlegungen qualitativ verglichen. Dabei wurden die Vorteile der hier entwickelten Interaktionstechniken insbesondere im Hinblick auf den Anwendungskontext 3D-Modellierung deutlich. Ein quantitaliver Vergleich mit kommerziellen Systemen im Rahmen eines Benutzertest hat ergeben, daß mit einem Modellierungssystem, das diese neuartigen Interaktionstechniken realisiert, ein Geschwindigkeitsvorteil von einem Faktor 2 bis 4 in der geometrischen Modellierung erzielt werden kann. In einem weiteren Versuch wurde

die Wichtigkeit der Systemunterstützung bei der interaktiven Zusammenbausimulation unterstrichen.

Mit einem Ansatz zur kombinierten 2D- und 3D-Cursor-Kontrolle in einem 3D-Eingabegerät konnte gezeigt werden, daß es prinzipiell möglich ist, auch die 2D-Benutzungsoberfläche mit einem 3D-Gerät zu bedienen und somit den Wechsel zwischen 3D- und 2D-Eingabegerät zu vermeiden. Der kombinierte Ansatz erhält die Vorteile der 3D-Interaktion durch kontext-sensitives Umschalten. Die durchgeführten Experimente haben gezeigt, daß diese Kombination Vorteile insbesondere für im Umgang mit 3D-Eingabegeräten unerfahrene Benutzer bietet.

Schließlich wurden verschiedene visuelle Hinweise auf ihren Einfluß auf Wahrnehmung und Interaktion evaluiert, besonderes Augenmerk galt dabei der stereoskopischen Ausgabe, der Rolle von Schattenwurf und deren Wechselwirkungen. Im Hinblick auf diesen Untersuchungsfokus ist interessant, daß sich die Kombination von Schattenprojektion auf virtuelle Wände und stereoskopische Darstellung negativ auswirkt und die Interaktionsgeschwindigkeit herabsetzt. Das Wahrnehmungssystem des Betrachters kann die beiden virtuellen Tiefeninformationen nicht plausibel in Einklang bringen.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

Den Schwerpunkt der vorliegenden Arbeit bildet die Konzeptionierung, Realisierung und Evaluierung neuartiger Verfahren und Techniken zur graphischen Interaktion mit 3D-Eingabegeräten für die effiziente und präzise Volumen- und Baugruppenmodellierung sowie die Erarbeitung unterstützender Visualisierungstechniken, die dem Benutzer aufschlußreiche Form- und Tiefenhinweise bieten und ihm die im Gange befindliche Interaktion geeignet darstellen.

Umgesetzt und eingebettet sind diese Techniken in dem benutzer-zentrierten Modellierungssystem ARCADE, für das basierend auf den allgemeinen Anforderungen an benutzer-zentrierte Systeme und den spezifischen Anforderungen von CAD zunächst ein Architekturkonzept erarbeitet wurde. Die entwickelte Systemarchitektur für benutzer-zentrierte Modellierungssysteme trägt den Anforderungen von CAD, Virtueller Realität und dem Kommunikationsbedarf verteilt arbeitender Mitarbeiter von Produktentwicklungsteams Rechnung.

Das Ziel der schnellen und präzisen 3D-Modellierung mit 3D-Eingabegeräten setzen die entwikkelten 3D-Interaktionstechniken durch Diskretisierung weitgehend unabhängig von der Präzision des verwendeten 3D-Eingabegerätes um. Auch wenn ARCADE, wie von benutzer-zentrierten Systemen gefordert, ein gewisses Spektrum an Ein- und Ausgabemöglichkeiten unterstützt (z.B. Monitor, Virtueller Tisch, Großbildprojektion), stand der herkömmliche CAD-Arbeitsplatz erweitert um die Möglichkeit zur 3D-Ein- und -Ausgabe im Mittelpunkt des Interesses, insbesondere was die Evaluierungen anbelangt. Vorrangiges Ziel war es, durch geeignete Interaktionstechniken 3D-Eingabe für den Konstrukteur an dem gewohnten Arbeitsplatz nutzbringend einsetzbar zu machen.

Dazu wurden folgene Schritte bearbeitet:

- Erhebung der Anforderungen der Modellierung an ein 3D-Eingabegerät für den herkömmlichen Arbeitsplatz.
- Vergleich und Bewertung verschiedener 3D-Eingabegeräte.
- Erhebung der Anforderungen der Modellierung an die Geometrieerzeugung und -manipulation.
- Konzeptionierung und Realisierung von ein- und zweihändigen 3D-Interaktionstechniken zur Objekterzeugung und -modifikation bei der Bauteilmodellierung und Zusammenbausimulation.

Die erarbeiteten 3D-Interaktionstechniken sind in mehreren Zyklen evaluiert und optimiert worden. Sie zeichnen sich durch ein hohes Maß an Intuitivität aus und setzen das Prinzip der Reiz-Reaktions-Korrespondenz für die Modellierung in 3D um. Die wichtigsten Techniken und Ergebnisse werden im weiteren zusammenfassend dargestellt.

Mit der Topologie-basierten eingeschränkten Modifikationstechnik (TCBM) wurde ein neues Interaktionsparadigma geschaffen, welches die Freiheitsgrade von 3D-Eingabegeräten auf vorteilhafte Weise nutzt, um gesten-basiert Manipulationen mit reduzierter Anzahl von Freiheitsgraden abzuleiten und durchzuführen, wie sie in der Modellierung häufig benötigt werden. Anhand der Geste können verschiedene Manipulationsformen ohne Moduswechsel erkannt werden. Dies minimiert die Anzahl der nötigen Interaktionen und reduziert die Komplexität der Benutzungsoberfläche. Die Reiz-Reaktions-Korrespondenz in 3D bleibt gewahrt. Die TCBM kann ein- oder auch zweihändig ausgeübt werden.

Zur Objekterzeugung wurden semantisch-korrekte, typbezogene Methoden entwickelt, die die Parameter von 3D-Primitiven nicht aus einer BoundingBox ableiten, die durch den Start- und Endpunkt der Interaktion definiert ist, sondern die Parameter aus einem Gestenvektor unter Beachtung objektspezifischer Regeln berechnen. Im Unterschied zu dem BoundingBox-basierten Verfahren kann bei diesem Ansatz die Objektposition über die Erzeugung invariant gehalten werden, was das genaue Positionieren und Erzeugen insbesondere von Objekten auf bereits existierenden Objekten erst ermöglicht.

Die Kombination dieser neuartigen Interaktionstechniken mit vielfältigen, benutzer-definierten Diskretisierungsmöglichkeiten erlaubt es erstmals, 3D-Eingabe in einem rein graphisch-interaktiven Dialog zur präzisen und effizienten Konstruktion einzusetzen. Ein Vergleich mit kommerziellen Systemen im Rahmen eines Benutzertests hat ergeben, daß sich in der präzisen Bauteilmodellierung ein Geschwindigkeitsvorteil von einem Faktor 2 bis 4 realisieren läßt. Damit sind die Mankos der aus der Literatur bekannten Verfahren aufgehoben, die 3D-Eingabe bislang nur für unpräzises Arbeiten erschliessen konnten. Für die Eingabe präziser Werte wurde bisher z.B. auf Tastatur- bzw. Spracheingaben oder einen *valuator* zurückgegriffen. Ergänzt werden diese Interaktionstechniken durch direkt-manipulative Modellieroperationen, wie implizite Boolsche Operationen und direkt-manipulatives Sweeping, die den Konstruktionsprozeß weiter beschleunigen.

Neben präzisen Interaktionstechniken für die Volumenmodellierung wurden Skizzierungstechniken für die Freiformflächenerzeugung durch Handbewegungen an einem Virtuellen Tisch entwickelt. Durch Interpretation der Eingabedaten, die die vom Benutzer frei im Raum skizzierten Kurven in die von den Interpolationsmethoden gewünschte Form bringt, wird die Flexibilität des Benutzers bei der Eingabe gesteigert. Weiterhin wird er von explizitem Wissen über die Vorgehensweise der Interpolationsalgorithmen entbunden. Die Methoden zur Freiformflächenerzeugung beschränken sich bislang auf das unpräzise Skizzieren. Ein Ziel weiterführender Arbeiten ist daher, die für die Volumenmodellierung realisierte Präzision im Interaktionsprozeß durch Entwicklung adäquater Verfahren auf die Freiformflächenmodellierung zu übertragen.

Als essentiell für den Konstruktionsprozeß sowie das schnelle Navigieren und präzise Interagieren hat sich die Notwendigkeit eines effizienten Pickens und des Snappings des graphischen Echos auf die Oberfläche sowie die topologischen Elemente von 3D-Modellen erwiesen. Das Verfahren für Picking und insbesondere Snapping - also der Effekt, daß das graphische Echo automatisch von der Objektoberfläche angezogen wird, wenn es in deren Nähe ist - muß hinreichend schnell sein, sonst behindert es die Interaktion, anstatt den Benutzer zu unterstützen. Basierend auf dieser Erkenntnis wurde ein Verfahren zum schnellen Picking und Snapping des graphischen Echos auf die präzise Geometrie des CAD-Modells entwickelt. Beim Snapping macht sich der Algorithmus die Kohärenz zwischen sukzessiven Benutzeraktionen zunutze. Hierarchische Bounding-Box-Tests und das Konzept der ’neutralen Zone’ verhelfen dem Verfahren zu einem von der Komplexität der Szene weitgehend unabhängigen, konstanten Laufzeitverhalten.

Zur Unterstützung der Zusammenbausimulation wurde eine Assembling Assistance-Methode konzipiert und prototypisch umgesetzt. Im Unterschied zu der aus VR-Systemen bekannten Kollisionserkennung basierend auf facettierten Modellen, arbeitet die Assembling Assstance-Methode attribut-basiert auf der präzisen Geometrie des CAD-Modells. Wo Kollisionserkennung dem Benutzer nur Hinweise geben kann, daß zwei Bauteile während des interaktiven Zusammenbauversuchs sich durchdringen, geht der hier entwickelte Ansatz einen Schritt weiter: Er unterstützt den Benutzer bei der interaktiven Zusammenbausimulation, indem er mögliche Paßstellen erkennt, das einzubauende Teil automatisch ausrichtet und die Bewegungsfreiheitsgrade für weitere Interaktionen einschränkt. Ob die Einbaulage erreicht ist, kann ebenfalls mit Hilfe der Attribute festgestellt werden. Das Verfahren arbeitet lokal und weist den Nachteil auf, nicht sämtliche während des Einbauvorgangs auftretenden Kollisionen erkennen zu können.

Im Hinblick auf die Visualisierung stellte sich die Frage, welche Darstellungstechniken und -hilfen eine effiziente 3D-Interaktion unterstützen. Zu diesem Zweck wurde der Einfluß von stereoskopischer Darstellung und Schattenwurf evaluiert. Es kann festgestellt werden, daß indirekte 3D-Interaktionen[1] mit 3D-Eingabegeräten, wie der SpaceMouse, am konventionellen Arbeitsplatz bei monoskopischer Darstellung unter Zuhilfenahme von Visualisierungstechniken, wie virtuellen Wänden, Schatten und Transparenz, effizient ausgeführt werden können. Interessant war die Beobachtung, daß eine Kombination von Stereo- und Schatten-Projektion sich negativ auswirken kann und beim Betrachter die Tiefenwahrnehmung eher stört als sie günstig zu beeinflussen. Darüber hinaus wurden Visualisierungstechniken erarbeitet, die dem Benutzer die stattfindende Interaktion sowie die Diskretisierung und die erlaubten Modifikationsmöglichkeiten während einer parametrisierten Interaktion darstellen. Diese Visualisierungshilfen wurden in Benutzertests von der Mehrheit der Testpersonen als sehr gut und wünschenswert auch für das ihnen vertraute CAD-System beurteilt. Interaktion und Visualisierung muß stets gemeinsam betrachtet werden, da sie sich synergetisch beeinflussen und gemeinsam das Maß an Effizienz im Umgang mit einem System bestimmen.

Abschließend sei nochmals auf das entwickelte Architekturkonzept einer benutzer-zentrierten Modellierungsumgebung zurückgekommen, das Aspekte der Virtuellen Realität und Kooperation (CSCW - computer supported collaborative work) als integralen Bestandteil enthält. Das Konzept wurde in Form des Systems ARCADE implementiert und somit die Machbarkeit eines solchen integrierten Ansatzes gezeigt. Das System wurde als erweiterbare Umgebung ausgelegt. Beispielhafte Erweiterungen sind die Integration des feature-basierten, parametrischen Modellierers Sinfonia [110], des ’sketching tools’ Jot [62] und Virtual Building Life Cycle (VBLC), einem Ansatz zum 4D-Architektur-CAD [95].

Drei Komponenten dieser Systemarchitektur sind von besonderer Bedeutung:

- Der GraphicManager, der die 3D-Interaktions- und Visualisierungstechniken zur Verfügung stellt und mit Modellierungsfunktionalität angereichert wurde, um der Anforderung nach Direkt-Manipulation zu begegnen.
- Der HistoryManager, der den HistoryGraphen verwaltet. Der HistoryGraph ist eine Datenstruktur, die den bekannten CSG-Baum (constructive solid geometry) hinsichtlich Flexibilität und Funktionalität erweitert. Der HistoryGraph bildet auch die Basis für Interaktionen, die es dem Benutzer gestatten, komplexe Modelle basierend auf der transparenten Änderung der Primitive in ihrer Konstruktionshistorie zu modifizieren.

1. Bei der indirekten Interaktion befindet sich die Hand des Benutzers nicht an der Stelle des graphischen Echos.

- Der NetworkManager, der die CSCW-Funktionalität realisiert und den Austausch von 3D-Graphikdaten mit dem Austausch von Nachrichten auf semantischer Ebene kombiniert, um Echtzeitvisualisierung der Interaktionen entfernter Benutzer im kooperativen Betrieb, Telepräsenz und Eignung des Systems für schmalbandige Netze zu erreichen. Mit diesem neuartigen kombinierten Kommunikationsansatz können örtlich verteilte Benutzer simultan an verschiedenen Objekten in einem gemeinsamen virtuellen Konstruktionsraum arbeiten, während sie die Interaktionen der Partner quasi in Echtzeit mitverfolgen.

6.2 Ausblick auf weiterführende Arbeiten

Wie bereits erwähnt, liegen bei drei der im Rahmen dieser Arbeit entwickelten Ansätze Weiterentwicklungen nahe: Bei der semantisch parametrisierbaren Interaktion, bei der Assembling Assistance-Methode und bei der Freiformflächenmodellierung.

Die parametrisierbare Interaktion dient zur durch die Semantik bestimmten Einschränkung von Modifikationsmöglichkeiten während der Interaktion. Das in Kapitel 4.2.9.3 beschriebene Verfahren weist Beschränkungen hinsichtlich der Komplexität erlaubter Modifikationsbereiche auf. Solche Modifikationsbereiche können im Verlauf einer Konstruktion die Form von getrimmten Freiformflächen oder beliebigen Teilräumen von R^3 einnehmen. Eine entsprechende Erweiterung der parametrisierbaren Interaktionstechniken ist daher angebracht. Art und Umfang der verarbeiteten Semantik liegt dabei bei dem zugrundeliegenden semantischen Modellierer, der die Information generiert, mit der die Interaktion parametrisiert wird.

Die vorgestellte attribut-basierte Assembling Assistance-Methode wurde bislang nur für zylindrische Körper implementiert, daher sollte eine Komplettierung der Implementierung zum Nachweis der Umsetzbarkeit des Konzeptes auch für andere Körper erfolgen. Wichtiger jedoch ist die Kombination von Assembling Assistance und Kollisionserkennung, da der größte Vorteil der Assembling Assistance-Methode, nämlich präzise aber lokal zu arbeiten, zugleich auch ihr Nachteil ist: Kollisionen zwischen Bauteilen werden nur für potentiell korrelierende Attribute innerhalb des lokalen Betrachtungsbereiches erkannt. Die Kombination mit einer Kollisionserkennung kann hier Abhilfe schaffen.

Während Präzision für den Interaktionsprozeß mit 3D-Eingabegeräten für Volumenmodelle durch die im Rahmen dieser Arbeit entwickelten Interaktionstechniken realisiert werden konnte, gilt es, zukünftig adäquate Verfahren für die intuitive Freiformflächenmodellierung zu entwickeln. Dazu bieten sich semi-immersive Umgebungen, wie der Virtuelle Tisch, und direkte 3D-Eingabe mit fliegenden Eingabegeräte geradezu an, schaffen aber auch neue Probleme bei der präzisen Positionsbestimmung, da die heutige Trackingtechnologie nicht die Präzision eines tisch-gebundenen Eingabegerätes, wie der Space-Mouse, bietet. Neben der Präzision sind intuitive und echtzeitfähige Verfahren zur Flächenmanipulation zu entwickeln bzw. weiterzuentwickeln. Vorstellbar ist eine gesten-basierte Modifikation, bei der aus der Geschwindigkeit der Geste der Einzugsbereich für die Modifikation nach dem Prinzip von Impuls und Masse bestimmt wird; schnellere Bewegungen führen zu einer lokal beschränkteren Modifikation als langsamere. Der Einsatz von Tablett und Stift als Eingabegeräte legt die Entwicklung neuartiger ein- und zweihändiger Selektions- und Modifikationsmöglichkeiten für Freiformflächen nahe. Benutzertests sollten auch diese zukünftigen Entwicklungen begleiten.

Die Topologie-basierte Modifikationstechnik (TCBM) bietet sich als neuartiges Interaktionsparadigma für die Kombination mit einem Force-Feedback Ein-/Ausgabegerät an. Die Kraftrückkopplung ist in der Lage, den Benutzer die Objekte in der Szene spüren zu lassen. Kommt es nun

zu einer Modifikation mittels der TCBM, können durch entsprechende Steuerung des Kraftrückkopplungsgerätes die reduzierten Freiheitsgrade dem Benutzer spürbar vermittelt werden. Der Einsatz eines Kraftrückkopplungsgerätes verspricht somit eine weitere Steigerung an Intuitivität und Effizienz. Es lassen sich allerdings auch erhöhte Akzeptanzprobleme seitens der Benutzer erwarten, wenn diese mit einem roboterartigen Eingabegerät konfrontiert werden. Neben der TCBM eignen sich auch Snapping und parametrisierte Interaktionen für die Kraftrückkopplung. Die Kraftrückkopplung stellt eine scheinbar ideale Ergänzung der entwickelten Interaktionstechniken dar, was allerdings durch eine Evaluierung noch nachgewiesen werden müßte.

Generell bleibt festzustellen, daß die Weiterentwicklung der Hardware, insbesondere der Ein- und Ausgabegeräte, immer wieder neue Möglichkeiten eröffnet, um natürlicher in virtuellen Umgebungen zu arbeiten, wobei die benötigten Interaktionstechniken nicht im selben Maße variieren, sondern weitgehend durch den Applikationskontext bestimmt sind. In erster Linie sind hier kabellose 3D-Eingabegeräte und leichte kopfgebundene Projektionssysteme zu nennen. In Anbetracht der raschen Innovationen könnte bald nicht nur die Vision von kommerziellen, benutzer-zentrierten 3D-CAD-Systemen, die ein- und ausgabeseitig diesen Namen auch verdienen, sondern auch die von der alltäglichen 3D-Kommunikation, die unsere traditionellen 2D-Techniken, wie Papier und Bleistift ersetzt, Realität werden.

Literaturverzeichnis

1. 3SPACE User's Manual, Polhemus, 1993.
2. Abeln, O. (Hrsg.): "CAD-Referenzmodell - zur arbeitsgerechten Gestaltung zukünftiger computergestützter Konstruktionsarbeit", Teubner Verlag, Stuttgart, 1995.
3. Abeln, O. (Hrsg.): „Innovationspotentiale in der Produktentwicklung - Das CAD-Referenzmodell in der Praxis", Teubner Verlag, Stuttgart, 1997.
4. Aderiu, E., Quester, R.: "A Hierarchical Communication System for Distributed Concurrent Engineering", Proc. of the Integrated Broadband Communication for Automotive Industry Workshop, May, 1995.
5. Anantha, R., Kramer, G.A., Crawford, R.H.: „Assembly modelling by geometric constraint satisfaction", Computer-Aided Design, Vol. 28, No. 9, 1996.
6. Anderl, R.: „STEP - Grundlagen, Entwurfsprinzipien und Aufbau". Tagungsband CAD'92 - Neue Konzepte zur Realisierung anwendungsorientierter CAD-Systeme, 14. - 15. Mai, Berlin, Springer-Verlag, 1992.
7. Anderl, R.: „Produktdatentechnologie III - Produktdatenmanagement". Skriptum zur Vorlesung an der Technischen Hochschule Darmstadt, Fachgebiet Datenverarbeitung in der Konstruktion (DiK), 1996.
8. Anderl, R., Bumiller, J., Momberg, M., Schiemenz, K., Schmidt, K., Stupperich, M.: „Multimediale Unterstützung verteilter Produktentwicklung". Tagungsband CAD'98, pp. 3 – 12, Hrsg. Prof. Anderl, Prof Encarnação, Informatik Xpress 9, 1998.
9. Anderl, R., Momberg, M.: „Authentication of STEP Product Models". Tagungsband ProSTEP Science Days '98, pp. 16 - 35, Hrsg. Prof. Anderl, Hr. Trippner, Informatik Xpress 10, 1998.
10. Astheimer, P., Böhm, K., Felger, W., Göbel, M., Müller, S.: Die Virtuelle Umgebung - Eine neue Epoche in der Mensch-Maschine-Kommunikation. Teil I: Einordnung, Begriffe und Geräte, Informatik-Spektrum, pp. 281 - 290, Springer Verlag, 1994.
11. Astheimer P., Böhm K., Felger W., Göbel M., Müller S. : „Die Virtuelle Umgebung - Eine neue Epoche in der Mensch-Maschine Kommunikation. Teil 2", Informatik Spektrum, pp 357 - 367, Springer Verlag, 1994.
12. Ayatsuka, Y., Matsuoka, S., Rekimoto, J.: „Penumbrae for 3D Interaction", Proc. ACM UIST '96, pp. 165 - 166, Seattle, Washington, 1996.
13. Azuma, R. T.: „A Survey of Augmented Reality", in Presence: Teleoperations and Virtual Environments 6, 4 , pp. 355 - 385, August 1997.
14. Barco Baron Product Flyer, http://www.barco.com, Barco 1997.

15. Bier, E.A., Stone, M.: „Snap-Dragging“, Proceedings SIGGRAPH 86, pp 233 - 240, ACM Press, 1986.

16. Bier, E.A.: „Skitters and jacks: interactive 3D positioning tools“, Proceedings of the 1986 workshop on interactive 3D graphics, pages 183-196, Chapel Hill, October 1986.

17. Bier, E.A.: „Snap-Dragging in Three Dimensions“, Proceedings of ACM Symposium on Interactive 3D Graphics, pp. 193 - 204, ACM Press, 1990.

18. Bier, Eric A.; Stone, Maureen C.; Fishkin, Kenn; Buxton, William; Baudel, Thomas: „A Taxonomy of See- Through Tools“, Proceedings of CHI ‘94, pp. 358- 364, ACM Press 1994.

19. Bier, Eric A.; Stone, Maureen C.; Pier, Ken; Buxton, William; DeRose, Tony D.: „Toolglasses and Magic Lenses: The See-Through Interface“, Proceedings of Siggraph ‘93 (Anaheim, August), Computer Graphics Annual Conference Series, pp.73- 80, ACM Press 1993.

20. Bimber, O.: „Rudiments of a 3D freehand sketch based human-computer interface for immersive virtual environments“. Proceedings of Virtual Reality Systems and Technology (VRST'99), pp. 182-183, 1999.

21. Bimber, O.: „Multimodal Object Interaction in Virtual Environments“. Technical Report, no. 99i001-FEGD, Fraunhofer Institute for Computer Graphics, 1999.

22. Blinn, J.: „Me and my fake shadows“. in Digest of Jim Blinn's Corner, Volume I, A trip down the graphics pipeline, Morgan Kaufmann Publishers, Inc., 1996.

23. Böhm, K., Hübner, W., Väänänen, K.: „GIVEN: Gesture Driven Interactions in Virtual Environments. A Toolkit Approach to 3D Interactions“, Proc Informatique 92: Interface to Real and Virtual Worlds, pp. 243 - 254, 1992.

24. Böhm, K.: „Ein generisches 3D-User Interface Toolkit mit Verfahren zur Gebärdenerkennung“. Dissertation, TH Darmstadt, 1996.

25. Böhm, T.: „Bidirektionale Abbildung von STEP-basierten geometrischen / topologischen Informationen aus einer objekt-orientierten Datenbank auf Datenstrukturen eines geometrischen Modellierers", Diplomarbeit, Fachhochschule Darmstadt, 1995.

26. Brunetti, G. and Vieira, A.S.: „Representation Scheme for Feature-based Parametric Design“, in Springer Book "CAD Systems Development - Tools and Methods", Eds. D. Roller und P. Brunet, pp. 165-179, Springer 1997.

27. Brunetti, G., De Martino, T., Falcidieno, B., Haßinger, S.: „A Relational Model for Interactive Manipulation of Form Features based on Algebraic Geometry“, Proceedings of the Solid Modeling'95, Third ACM/IEEE Symposium on Solid Modeling and Applications, Salt Lake City, Utah, 1995.

28. Brunetti, G., Stork, A.: ”Product-centred Intuitive 3D Interaction for Feature-based Parametric Assembly Modelling”, Tagungsband ProSTEP Science Days ‘98, pp. 61 – 76, Hrsg. Prof. Anderl, Hr. Trippner, Informatik Xpress 10, 1998.

29. Burdea, G.: „Force and Touch Feedback for Virtual Reality“, Wiley Interscience, 1996.

30. Butterworth, J., Davidson, A., Hench, S., Olano, T.M.: „3DM: A three dimensional modeler using a head-mounted display“, Communications of the ACM, pp 55-62, 1992.

31. Buxton, William; Myers, Brad A.: „A study in two-handed input“, Proceedings of CHI ‘86, Human Factors in Computing Systems, pp. 321- 326, ACM Press 1986.

32. Buxton, W. A.: „Directory of Sources to Input Technology“, 1998, http://www.dgp.utoronto.ca/people/BillBuxton/InputSources.html.

33. Carlsson, C., Hagsand, O.: „DIVE: A Multi User Virtual Reality System“, In Proceeding of IEEE 1993 Virtual Reality Annual International Symposium, pp. 394, 1993.

34. Chen, M., Mountford, S.J. and Sellen, A.: „A study in interactive 3-D rotation using 2-D control devices“, Computer Graphics (Siggraph’88 Proceedings), 22(4):121-129, August 1988.

35. Chin, N., Feiner, S.K.: „Near real-time shadow generation using BSP-Trees“, Computer Graphics (SIGGRAPH'89 Proceedings), Vol. 23(3), ACM Press, pp. 99 – 106, 1989.

36. Chu, Chi-Cheng P.; Dani, Tushar H,; Gadh, Rajit: „Multi-sensory user-interface for a virtual-reality-based Computer aided design system“, Computer-Aided Design, Vol. 29, No. 10, pp. 709-725, 1997.

37. Cohen, J., Lin, M., Manocha, D., Ponamgi, K.: „I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scaled Environments“. Proc. ACM Int. 3D Graphics Conference, pp. 189 - 196, 1995.

38. Cohen, J.M., Markosian, L., Zeleznik, R.C, Hughes, J.F. and Barzel, R.: „An Interface for Sketching 3D Curves“. 1999 Symposium on Interactive 3D Graphics, pp. 17-21, ACM 1999.

39. Conner, B., Cutts, M., Fish, R., Fuchs, H., Holden, L., Jacobs, M., Loss, B., Markosian, L., Riesenfeld, R., Turk, G.: „An Immersive Tool for Wide-Area Collaborative Design“. Proceedings TeamCAD: GVU/NIST Workshop on Collaborative Design, 1997.

40. Coons, S.A.: Surfaces for computer-aided design of space forms. Technical Report, MIT, 1967.

41. Coquillart, S.: „Extended Free-Form Deformation: A Sculpting Tool for 3D Geometric Modeling“. ACM Computer Graphics (Proc. of SIGGRAPH '90), 187-196, 1990.

42. Cruz-Neira, C., Sandin, D. J., DeFanti, T. A.: „Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE“. ACM SIGGRAPH’93 Conference Proceedings, Anaheim, CA, July 1993.

43. Cutler, L. D.; Fröhlich, B.; Hanrahan, P.: „Two-Handed Direct Manipulation on the Responsive Workbench“, Proc. Symposium on Interactive 3D Graphics, pp. 107-114, ACM Press, 1997.

44. Dani, T. H., Gadh, R.: „COVIRDS: A Conceptual Virtual Design System“, Computer Aided Design, Vol. 29(8):555-563, 1997.

45. Dani, T. H., Wang, L., Gadh, R.: „Free-Form Surface Design in a Virtual Environment“. Proc. of ASME '99 Design Engineering Technical Conferences, ASME 1999.

46. DIN 323 T1: „Normzahlen und Normzahlenreihen; Hauptwerte, Genauwerte, Rundwerte“. August 1974.

47. DIN 66 234 Teil 8: „Bildschirmarbeitsplätze - Grundsätze der ergonomischen Dialoggestaltung“. Beuth, Berlin, 1988.

48. Dietrich, U., v. Lukas, U., Morche, I.: “Kooperatives Modellieren auf der Basis von standardisierten Diensten”, Tagungsband CAD ’96 - Verteilte und intelligente CAD-Systeme”, S. 281 ff., DFKI GmbH, Kaiserslautern, März 1996.

49. Dietrich, U., von Lukas, U., Morche, I.: „Cooperative modelling with TOBACO“. Proceedings TeamCAD GVU/NIST Workshop on Collaborative Design, Atlanta, pp. 115 – 122, 1997.

50. Edmonds, E.: The Separable User Inerface. Academic Press, 1992.

51. Encarnação, J.L.: „Anno 2010 - Remembering our Future: Challenges and Frontiers of Human Media Technology as the Kernel for Human Centered Computing“, Invited Speech, EUROGRAPHICS ’98, 1998.

52. Encarnação, M., Stork, A.: "Adaptionsmöglichkeiten in modernen CAD-Systemen: Bewertung, Konzeption und Realisierung", Tagungsband CAD '96 - Verteilte und intelligente CAD-Systeme, pp. 342 - 356, Hrsg. D. Ruhland, DFKI, Kaiserslautern, März 1996.

53. Encarnação, M., Stork, A., Schmalstieg, D., Bimber, O., Barton, R.: "The Virtual Table - a future CAD workspace", Proceeding of SME Computer Technology Solutions Conference (formerly Autofact), Detroit, SME 1999.

54. Encarnação, M., Bimber, O., Schmalstieg, D., Chandler, S.: "A Translucent Sketchpad for the Virtual Table Exploring Motion-based Gesture Recognition", Eurographics '99, 18(3), 1999.

55. Fa, M., Fernendo, T., Dew, P.M.: „Interactive Constraint-based Solid Modelling using Allowable Motion“. Proceedings of the 2nd ACM Solid Modelling ’93, Montreal, Canada, May, 1993.

56. FeatureM Benutzerhandbuch, strässle Informationssysteme AG, 1995.

57. Felger, W.: „Innovative Interaktionstechniken in der Visualisierung“, Dissertation TH Darmstadt, 1995.

58. Fischer, R., Vieira, A. S., Rix., J.: „Analysis-aided geometric design based on constrained parametric structures“, Geometric Modelling, 1994.

59. Foley, J., van Dam, A., Feiner, S., Hughes, H.: „Comuter Graphics - Principles and Practice“, Addison Wesley, 1990.

60. Fontana, M. , Giannini, F. , Meirana, M.: „A Free-form Feature Taxonomy“. Eurographics '99, 18(3), 1999.

61. Forrest, A. R.: „User Interfaces for Three-Dimensional Geometric Modelling“. Proc. Symposium on Interactive 3D Graphics, pp. 237 - 249, 1986.

62. Forsberg, A., J. LaViola, L. Markosian, R. Zeleznik: „Seamless Interaction in Virtual Reality“, IEEE Computer Graphics and Applications, 17(6):6-9, November/December 1997.

63. Forsberg, Andrew S.; Strauss, Paul S.; Zeleznik, Robert C.: „Two Pointer Input For 3D Interaction“, Proc. Symposium on Interactive 3D Graphics, pp. 115-120,192, ACM Press, 1997.

64. Forsberg, A.S., LaViola, J.J., Zeleznik, R.C.: „ErgoDesk: A Framework for Two- and Three-Dimensional Interaction at the ActiveDesk“. Proceedings of the Second International Immersive Projection Technology Workshop, Ames, Iowa, May 11-12, 1998.

65. Fuchs, H., Kedem, Z.M., Naylor, B.F.: „On Visible Surface Generation by a priori tree structure“, Computer Graphics (SIGGRAPH'80 Proceedings). ACM Press, 1980.

66. Galyean, T., Gold, M., Hsu, W., Kaufman, H., Stern, M.: „Manipulation of virtual three-dimensional objects using 2D input devices“, Brown University, User Interface Seminar, 1989.

67. Galyean, T. A., Hughes, J. F.: „Sculpting: An Interactive Volumetric Modelling Technique“, ACM Computer Graphics, 25(4), pp. 267 - 274, July 1991.

68. GI-Richtlinie: „Ergonomische Gestaltung der Benutzungsschnittstellen von CAD-Systemen“. GI-Empfehlung, 1996.

69. Gomes, A., Kress, H., Müller, S.: „Digital Mock-Up in der Einbau- und Montagesimulation“, Tagungsband Systems 97, 1997.

70. Gordon, W.: „Spline-blended surface interpolation through curve networks“. Journal of Math and Mechanics, 18(10):931-952, 1969.

71. Grissom, S.B., Perlman, G.: „StEP(3D): a standardized evaluation plan for three-dimensional interaction techniques“, Int. J. Human-Computer Studies 43, 15-41, 1995.

72. Hassinger, S.: „COSMOS - Construction and Modelling System, Implementation Guide Vers. 0“, Interner Abschlußbericht, Fraunhofer-IGD Darmstadt, 1994.

73. Herndon, K.P., Zeleznik, R.C., Robbins, D.C, Conner, D.B., Snibble, S.S., and van Dam, A.: „Interactive shadows“, UIST ‘92 Proceedings, pages 1-6, Nov. 1992.

74. Hollands, R.: „3D navigational & gestural devices“, VR News, Vol. 4 (2), Cydata Li mited, London, pp. 23 - 29, March 1995.

75. Hsu, W.M. , Hughes, J.F. , Kaufman, H.: „Direct Manipulation of Free-Form Deformations“. ACM Computer Graphics (Proc. of SIGGRAPH '92), 177-184, Chicago, July 26-31, 1992.

76. Hummels, C. C. M., Paalder, A. E., Overbeeke, C. J., Stappers, P. J., Smets, G.: „Two-Handed Gesture-Based Car Styling in a Virtual Environment“. Proceedings ISATA’97, pp. 227 - 234, Florence, Italy, June 16 - 19 1997.

77. Hummels, C., Overbeeke, C., Stappers, P.J., Kehler, T.: „Exploiting the Expressive: Rapid Entry of Car Designers' conceptual Sketches into a CAD Environment“. Proceedings ISATA’97, Florence, Italy, June 16 - 19 1997.

78. Igarashi, T., Matsuoka, S., Tanaka, H.: „Teddy: A Sketching Interface for 3D Freeform Design“. *ACM Computer Graphics (Proc. of SIGGRAPH ’99)*, ACM 1999.

79. ISO 10303-1: „Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles“, International Organization for Standardization, Geneva, Switzerland, 1994.

80. ISO DIS 10303-22: „Industrial automation systems and integration - Product data representation and exchange - Part 22: Implementation methods: Standard data access interface specification“, ISO TC184/SC4. National Institute of Standards and Technology, Gaithersburg, MD, USA, 1995.

81. Jasnoch, U., Haas, S.: „A Collaborative Environment based on Distributed Object-Oriented Databases“. Computers in Industy 29, pp. 51 - 61, Elsevier, 1996.

82. Jasnoch, U.: „Eine offene, verteilte CAD-System Umgebung zur Unterstützung von Concurrent Engineering - Konzept für einen Konsistenz-Manager“. Dissertation, TH Darmstadt, 1996.

83. Jasnoch, U., Anderson, B.: “Integration techniques for distributed visualization within a Virtual Prototyping Environment”, Proc. IS&T/SPIE Conference on Electronic Imaging, Science & Technology, 28.1. - 2.2.96, San Jose, USA, 1996.

84. Jayaram, S., Wang, Y., Jayaram, U., Lyons, K., Hart, P.: „A Virtual Assembly Design Environment“, Proceedings IEEE VR’99, IEEE 1999.

85. Julesz, B., Fritsch, H.L., Gilbert, E.L., Shepp, L.A.: „Inability of Humans to Discriminate between Visual Textures that Agree in Second-Order Statistics - Revisited“, Perception, Vol. 2, pp. 391 - 405, 1973.

86. Kjelldahl, L. and Prime, M.: „A study on how depth perception is affected by different presentation methods of 3D objects on a 2D display“, Computers & Graphics, Vol. 19, No. 2, pp. 199-202, 1995.

87. Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.P.: „Interactive Multi-Resolution Modeling on Arbitrary Meshes“. ACM Computer Graphics (Proc. of SIGGRAPH '98), 105-114, Orlando, Florida, July 19-24, 1998.

88. Koch M., Hassinger S.: „Konfigurierbare CAD-Applikationen auf der Basis eines flexiblen Benutzungsoberflächen-Systems“, in: Gausemeier Jürgen (Hrsg.), CAD’94 ”Produktdatenmodellierung und Prozessmodellierung als Grundlage neuer CAD-Systeme”, Fachtagung der Gesellschaft für Informatik e.V., Paderborn, 17./18.3.94, Hanser Verlag, pp. 330, 1994.

89. Kripac, J.: „A Mechanism for Persistently Naming Topological Entities in History-Based Parametric Solid Models“. Proc. ACM SIGGRAPH 3rd Symposium on Solid Modeling and Applications, pp. 21 - 30, May 15 - 17, 1995.

90. Krishnan, S., Gopi, M., Lin, M., Manocha, D., Pattekar, A.: „Rapid and Accurate Contact Determination between Spline Models using ShellTrees“. Eurographics ’98, 1998.

91. Krüger, W. and Fröhlich, B.: „The Responsive Workbench (virtual work environment)“, IEEE Computer Graphics and Applications, pp. 12 - 15, Volume 14, Issue 3, IEEE 1994.

92. Kuriyama, S.: „Surface modeling with an irregular network of curves via sweeping and blending“. Computer-Aided Design, 16(8):597-606, 1994.

93. Leganchuk, A., Zhai, S., Buxton, W.: „Bimanual Direct Manipulation in Area Sweeping Tasks“, URL: http://www.dgp.utoronto.ca/people/andrea/bimanual.html.

94. Liang, J.D. and Green, M.: „JDCAD: A highly interactive 3D modelling system“, Computers and Graphics, 18(4), pages 499-506, 1994.

95. Linnert, C., Stork, A., Encarnação, M., Koch, V.: „Virtual Building Life Cycle“, erscheint in Proceedings of 8th International Conference on Computing in Civil & Building Engineering, August 14 – 17, Stanford, California, 2000.

96. Liu, H. C., Srinath, M. D.: „A String Descriptor for Matching Partial Shapes“. In Shapiro, S., Rosenfield, A (eds.) Computer Vision and Image Processing, pp. 575 - 592, Academic Press, 1992.

97. Lüddemann, J.: „Virtuelle Tonmodellierung zur skizzierenden Formgestaltung im Industriedesign“. Dissertation TU Berlin, Berichte aus dem Produktionstechnischen Zentrum Berlin, UNZE Verlag, Potsdam 1996.

98. Magellan 3D Controller - User’s Manual Version 4.2 Logitech, 1995.

99. Maidhof, M.: „Konzeption und Realisierung geeigneter Interaktionstechniken für CAD-Systeme mittels eines 3D-Eingabegerätes“, Diplomarbeit FH Darmstadt, 1996.

100. Markosian, L., Cohen, J.M., Crulli, T., Hughes, J.F.: „Skin: A Constructive approach to Modelling Free-Frorm Shapes“, ACM Computer Graphics (Proc. SIGGRAPH '99), 1999.

101. Mine, M. R.: „Working in a virtual world: Interaction techniques used in the Chapel Hill immersive modelling program“, Technical Report 1996-029, 1996.

102. Müller, S., Kresse, W., Gatenby, N., Schöffel, F.: „A radiosity approach for the simulation of daylight“. In Proc. of the Sixth Eurographics Workshop on Rendering (Juni 1995) In: P. M. Hanrahan, W. Purgathofer (eds.): Rendering Techniques '95, pp. 137-146, Springer Verlag 1995.

103. Murakami, T.: „Direct and intuitive input device for 3D shape design“, DE-Vol. 83, 1995 Design Engineering Technical Conferences, Vol. 2, pp. 695-701, ASME, 1995.

104. Nawotki, A.: „Selective Crypting with Hair-Wavelets“. in „CAD tools and algorithms for product design“, Eds.: P. Brunet, C. Hoffmann, D. Roller., Springer Verlag, 2000.

105. Naylor, B. F.: „Sculpt: An interactive solid modeling tool“. Proceedings of Graphics Interface’90, pp. 138 - 148, May 1990.

106. Naylor, B.F., Amantides, J., Thibault, W.C.: „Merging BSP-Trees yields Polyhedral Set Operation“, Computer Graphics, Vol. 24(4), August, ACM Press, pp. 115 – 124, 1990.

107. Nielson, G.M. and Olson, D.R.: „Direct manipulation techniques for 3D objects using 2D locator devices“, Proceedings of the 1986 workshop on interactive 3D graphics, pages 174-182, Chapel Hill, October 1986.

108. Object Management Group: „The Common Object Request Broker Architecture“, John Wiley & Sons, Inc., 1992.

109. OpenGL Optimizer Programmer's Guide, Beta Draft (May 21), SGI, 1997.

110. Ovtcharova, J., Haßinger, S., Vieira, A.S., Jasnoch, U., Rix, J.: „Sinfonia, An Open Feature-based Design Module“, Proceedings of the 14th ASME International Computers in Engineering (CIE'94) Conference, Minnesota, 1994.

111. Pahl, G.: „Konstruieren mit 3D-CAD Systemen - Grundlagen, Arbeitstechnik, Anwendungen“, Springer-Verlag, 1990.

112. Pegasus 3D-CyberBat, TheOwl, Handbuch Edition 2, PEARL Agency Allgemeine Vermittlungsgesellschaft mbH, Buggingen 1995.

113. Piecha, R.: ”Zweihändige Eingabe in 3D-CAD unter Berücksichtigung der Anforderungen eines Virtuellen Tisches”, Diplomarbeit, Fachhochschule Darmstadt, 1998.

114. Piegl, L., Tiller, W.: The NURBS Book. Springer, 1997

115. Pique, M.: „Semantics of interactive Rotations“, Symposium of Interactive Graphics Proceedings ‘86, pages 259-269, Oktober 1986.

116. Preuss, J.: ”Binary Space Partitioning Trees zur direkt-manipulativen Volumenmodellierung”, Diplomarbeit, Fachhochschule Darmstadt, 1997.

117. Rappoport, A.: „Direct manipulation devices for the design of geometric constraint networks“. Magnenat-Thalmann, N., Thalmann, D. (eds), Communicating with Virtual Worlds, pp. 294-305, Springer Verlag, 1993.

118. Rappoport, A., Spitz, S.: „Interactive Boolean operations for conceptual design of 3-D solids“. Proceedings of Siggraph 97. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, pp. 269-278, 1997.

119. Reeves, W. T., Salesin, D. H., Cook, R. L.: „Rendering antialiased shadows with depth maps“. Proc. SIGGRAPH '87, volume 21, pages 283-291, July 1987.

120. Requicha, A. A. G.: „Representation of rigid solids: Theory, methods and systems“. ACM Computing Surveys 12(4), pp. 437 - 464, December 1980.

121. Requicha, A. A. G., Voelcker, H. B.: „Boolean Operations in Solid Modelling: Boundary Evaluation and Merging Algorithms“. Proc. IEEE, 73(1), pp. 30 - 44, January 1985.

122. Richter, T.: „Konzeptionierung und Realisierung einer Erweiterung eines 3D-Modellierers zum kooperativen Arbeiten in heterogenen Umgebungen“, Diplomarbeit FH Darmstadt, 1996.

123. Rock, I.: „Wahrnehmung: vom visuellen Reiz zum Sehen und Erkennen“, Spektrum der Wissenschaft, 1985.

124. Rosen, D. W., Bras, B., Mistree, F., Goel, A.: „Virtual Prototyping for Product Demanufacture and Service Using a Virtual Design Studio Approach“. Proceedings ASME Computers in Engineering Conference, pp. 951 - 958, Boston, 1995.

125. Rubine, D.: „Combining Gestures and Direct Manipulation“. Proc. CHI’92, pp. 659 - 660, May 3 -7, 1992.

126. Sachs, Emanuel; Roberts, Andrew; Stoops, David: „3-Draw: A Tool for Designing 3D Shapes“, IEEE Computer Graphics & Applications, pp. 18- 26, IEEE Computer Society Press, 1991.

127. Scharr, J., G. Brunetti, U. Müller: „Towards a Representation Scheme for Feature-based Parametric Assembly Modelling“, Proceedings of the 3Oth International Symposium on Automotive Technology & Automation, Florence, Italy, June l6th-l9th, pp.217-226, 1997.

128. Scharr, Jochen: „Konzeptionierung generischer Verbindungs-Features auf Basis der Anforderungen in der Betriebsmittelkonstruktion der Mercedes-Benz AG“, Diplomarbeit im Fachgebiet Datenverarbeitung in der Konstruktion, TH Darmstadt, 1996.

129. Schimpke, O.: „Freiformflächenmodellierung am Virtuellen Tisch“. Diplomarbeit Fachhochschule Darmstadt, 1999.

130. Schkolne, S., Schröder, P.: „Surface Drawing“. Technical Report CS-TR-99-03, Caltech Department of Computer Science,1999.

131. Schmalstieg, D., Encarnação, L.M., Szalavári, Z.: "Using Transparent Props For Interaction With The Virtual Table", Proc. Symposium on Interactive 3D Graphics '99, ACM 1999.

132. Schöffel, F.: „Online Radiosity in Interactive Virtual Reality Applications“. VRST '97 (Proceedings of the ACM Symposium on Virtual Reality Software and Technology), pp. 201 - 208, Sept. 1997.

133. Schultz, R., Jansen, H., Böttge, U.: „Praxisorientierte Telekooperationssysteme für Prozeßkettender verteilten Produktentwicklung“, Tagungsband CAD 2000 - Kommunikation, Kooperation, Koordination, Berlin März 2000, pp. 3-26, GI Bonn, 2000.

134. Schwert, W.: „Evaluierung und Optimierung von 3D-Interaktionstechniken im CAD“, Diplomarbeit FH Darmstadt, 1997.

135. Sederberg, T.W., Parry, S.R.: „Free-Form Deformations of Solid Geometric Models“. ACM Computer Graphics (Proc. of SIGGRAPH '86), 151-160, August 1986.

136. Segal, M., Korobkin, C., van Widenfelt, R., Foran, J., Haeberli, P. E.: „Fast shadows and lighting effects using texture mapping“. Proc. SIGGRAPH '92, volume 26, pages 249-252, July 1992.

137. Seibert, F., Kühner, H.: „A Vision based System controlling the Virtual Table's Camera“, in Virtual Reality for Industrial Applications. Fan Dai (Ed.), Springer-Verlag 1997.

138. Shah, J.J. and Rogers, M.T.: „Assembly modeling as an extension of feature-based design“, Res. Engineering Design, Vol. 5, pp. 218-237, 1993.

139. Shah, J.J. and Tadepalli, R.: „Feature-based assembly modeling“. Comput. Engineering. Vol. 1, 1992.

140. Shaw, C., Green, M.: „Two-Handed Polygonal Surface Design“, UIST 94, pp. 205-213, ACM Press, 1994.

141. Shneiderman, B.: „Direct Manipulation: a step beyond programming languages“, IEEE Computer, 16(8), pp. 57-69, 1983.

142. Showcase Online Manual, SGI, 1996.

143. Siemens Nixdorf Informationssysteme Produktinformationen zu jointX, http://www.sni.de/public/media/telecoop/telec_us/joint/jointx.htm, 1997.

144. Slater M., Davidson A.: „Liberation from Flatland: 3D Interaction Based on the Desktop Bat“, F.H. Post and W. Barth, editors, Eurographics ’91, pp. 209-221, Elsevier Science Publishers B.V. (North-Holland), 1991.

145. Spatial Technology Inc.: „ACIS Geometric Modeler Application Guide“, Boulder, Colorado, 1996.

146. Spur, G., Krause, F.-L.: „CAD-Technik: Lehr- und Arbeitsbuch für die Rechnerunterstützung und Arbeitsplanung“. Hanser, München, 1984.

147. Spur, G., Krause, F.-L.: „Das virtuelle Produkt - Management der CAD-Technik“. Hanser, München, 1997.

148. Steed, A. and Slater, M.: „3D interactions with the desktop bat“, Computer Graphics Forum, 14(2), 97-104, Blackwell, 1995.

149. Stork, A., Anderson, B.: "3D Interfaces in a Distributed Modelling Environment - 3D Devices, Interaction and Visualization Techniques", GI Workshop on Modelling, Virtual Reality and Distributed Graphics, MVD '95, 27. - 28. Nov. 1995, Bad Honnef / Bonn, pp. 83 - 92, Hrsg. D.W. Fellner, infix-Verlag, 1995.

150. Stork, A., Brunetti, G., Vieira, S.: "Intuitive Semantically Constrained Interaction in Feature Based Parametric Modelling", Tagungsband CAD '96 - Verteilte und intelligente CAD-Systeme, pp. 77 - 91, Hrsg. D. Ruhland, DFKI, Kaiserslautern, März 1996.

151. Stork, A., Jasnoch, U., J. Rix: ”Virtual Reality und Kooperation als integraler Bestandteil von CAD: Eine vereinheitlichte Systemarchitektur”, Tagungsband CAD’98, pp. 34 – 49, Hrsg. Prof. Anderl, Prof Encarnação, Informatik Xpress 9, 1998.

152. Stork, A., Jasnoch, U.: "A Collaborative Engineering Environment", Proceedings TeamCAD 1st GVU Workshop on Network-Centric CAD, pp. 25 - 33, Georgia, Altanta, 1997.

153. Stork, A., von Lukas, U., Schultz, R.: ”Enhancing a Commercial 3D CAD System by CSCW Functionality for Enabling Co-operative Modelling via WAN”, Proceedings ASME DECT’98, Atlanta, Georgia, 1998.

154. Stork, A., Maidhof, M.: „Efficient and Precise Solid Modelling using a 3D Input Device“, Proceedings of the ACM SIGGRAPH Symposium on Solid Modelling and Applications, May 14 – 16, Georgia, Atlanta, pp 181-194, 1997.

155. Stork, A., Richter, T., Quester, R.: "Collaborative Design with 3D Input Devices in a Heterogenous Distributed Environment", 1st Annual Conference on Applied Concurrent Engineering - ACE '96, November 5 - 7, Seattle, Washington, 1996.

156. Stork, A.: "Interfacing to a CSG History Graph", Proceedings of the CSG '96 Conference - Set-theoretic Solid Modeling: Techniques and Applications, pp. 201 - 214, Information Geometers, Winchester, April 1996.

157. Stork, A., von Lukas, U., Schultz, R., Widmer, H. J.: „Umsetzung der verteilten Konstruktion (WAN-Netze) im Bereich der Produktentwicklung“. in Verbundprojekt CAD-Referenzmodell Phase II - ergänzende Projektdokumentation, 1998.

158. Stork, A.: „An Algorithm for Fast Picking and Snapping in 3D using a Full Space Cursor and a 3D Input Device“. in „CAD tools and algorithms for product design“, Eds.: P. Brunet, C. Hoffmann, D. Roller., pp. 113-127, Springer Verlag, 2000.

159. Stork, A.: "Interaktive Zusammenbausimulation mittels implizit generierter Assembly-Attribute", Tagungsband CAD 2000 - Kommunikation, Kooperation, Koordination, Berlin März 2000, pp. 437-452, GI Bonn, 2000.

160. Stork, A., de Amicis, R.: "ARCADE/VT - a Virtual Table-centric modeling system", Proceedings of 4th International Immersive Projection Technology Workshop (IPT 2000), June 19-20, Ames, Iowa, 2000.

161. Stork, A., Schimpke, O., de Amicis, R.: „Sketching Freeforms in semi-immersive virtual environments“, erscheint in Proc. ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2000), Baltimore, Maryland, September 10-13, 2000.

162. Sun Solutions: The complete Guide to ShowMe 2.0.1., 1994.

163. Szeliski, R., Tonnesen, D.: „Surface Modeling with Oriented Particle Systems“. ACM Computer Graph-ics (Proc. of SIGGRAPH '92), 185-194, Chicago, July 26-31, 1992.

164. Turner, R., Gobbetti, E., Soboroff, I.: „Head-Tracked Stereo Viewing with Two-Handed 3D Interaction for Animated Character Construction“, EUROGRAPHICS ‘96 pp. C-197 - C-206, Eurographics Association, 1996.

165. Turner, A., Chapman, D. and Penn, A.: „Sketching a Virtual Environment: Modeling Using Line-Drawing Interpretation“. Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST'99), pp. 155-161, 1999.

166. Vance, J. M., Furlong, T. J., Larochelle, P. M.: „Spherical Mechanism Synthesis in Virtual Reality“. Proceedings ASME DETC’98, Atlanta, Georgia, 1998.

167. van Emmerik, M.: „A Direct Manipulation Technique for Specifying 3D Object Transformations with a 2D Input Device“, Computer Graphics Forum No. 9. pages 355 - 361, North-Holland, 1990.

168. VDI 2249E : „CAD-Benutzungsfunktionen“. VDI, 1999.

169. Vieira, A. S.: „Consistency management in feature-based parametric design“, Computer-Integrated Concurrent Design, Symposium of the Design Technical Conferences of the American Society of Mechanical Engineers ASME, Boston, MA, September 17-21, 1995.

170. von Lukas, U., Krautstein, T., Schultz, R., Stork, A., Widmer, H.-J.: „Einfuhrung von Telekooperationstechniken in der Produktentwicklung”, Tagungsband CAD’98, pp. 306 – 315, Hrsg. Prof. Anderl, Prof Encarnação, Informatik Xpress 9, 1998.

171. von Lukas, U., Stork, A.: „Introducing Tools for Collaborative Modelling“, Proceedings ISATA '98, 1998.

172. von Lukas, U., Mähler, A., Scheinhof, E.: „Kommunikation und Kooperation in der integrierten verteilten Produktentstehung“, Tagungsband CAD 2000 - Kommunikation, Kooperation, Koordination, Hrsg. A. Iwainsky, pp. 27 - 50, 2. und 3. März, Berlin, GI 2000.

173. Wanger, L.R., Ferwerda, J.A. and Greenberg, D.P.: „Perceiving spatial relationships in computer-generated images“, IEEE Computer Graphics and Applications, 12(3):44-58, May 1992.

174. Ware, C. and Jessome, D.R.: „Using the bat: a six -dimensional mouse for object placement“, IEEE Computer Graphics and Application, 11, pages 65-70, 1988.

175. Ware, C., Osbourne, S.: „Exploration and Virtual Camera Control in Virtual Three Dimensional Environments“. ACM Computer Graphics, 24(2), pp. 175 - 183, 1990.

176. Weber, C.: „What is a feature and what is its use? -Results of FEMEX Working Group I“, 29th International Symposium on Automotive Technology & Automation, Mechatronics - Advanced Development Methods and Systems for Automotive Products, Automotive Automation Limited 1996.

177. Welch, W., Witkin, A.: „Variational Surface Modeling“. ACM Computer Graphics (Proc. of SIGGRAPH '92), 157-166, 1992.

178. Wernecke, J.: “The Open Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor”, Addison Wesley, 1993.

179. Wiegand, T. F.: „Interactive Rendering of CSG Models“. Computer Graphics Forum 15(4), 1996.

180. Whitney, D. E.: „The Potential for Assembly Modeling in Product Development and Manufacturing“. MIT 1996.

181. www.sensable.com

182. www.virtex.com

183. Zachmann, G.: „Rapid Collision Detection by Dynamically Aligned DOP-Trees“, Proceedings of VRST 98, 1998.

184. Zeleznik, R.C., Herndon, K.P. and Hughes, J.F.: „SKETCH: An Interface for Sketching 3D Scenes“, Computer Graphics (Siggraph’96 Proceedings). pages 163-170, August 1996.

185. Zhai, S., Buxton, W., Milgram, P.: „The ’Silk Cursor’: Investigating Transparency for 3D Target Acquisition“, CHI’94 Proceedings, pages 459-464, April 1994.

186. Zheng, J.M., Chan, K.W., Gibson, I.: „Multiple-Points Constraints based Deformation for Free-Form Surfaces“. Proc. 1999 ASME Design Engineering Technical Conferences, September 12-15, Las Vegas, Nevada, ASME 1999.

Abbildungsverzeichnis